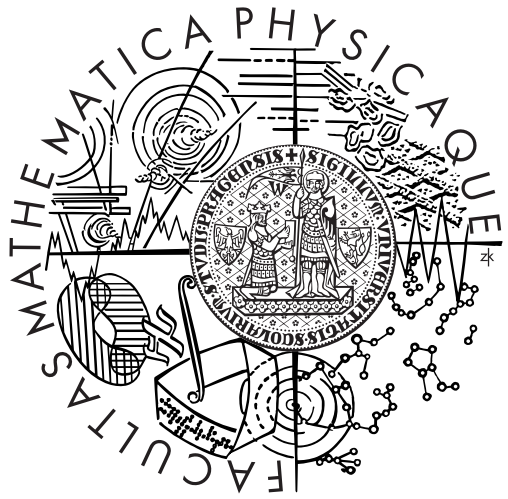
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Matej Dajčár

Implementace alternativních metrik v protokolu AODV

Katedra softwarového inženýrství
Vedoucí diplomové práce: Mgr. Miroslav Novotný

Studijní program: Informatika, softwarové systémy

Chcel by som poďakovať vedúcemu tejto diplomovej práce pánovi Mgr. Miroslavovi Novotnému za vedenie tejto práce, odborné postrehy a pripomienky, ktorými prispel k vypracovaniu práce.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 30.11.2010 Matej Dajčár

Obsah

Název práce: Implementace alternativních metrik v protocolu AODV
Autor: Matej Dajčár
Katedra: Katedra softwarového inženýrství
Vedoucí diplomové práce: Mgr. Miroslav Novotný
e-mail vedoucího: novotny@ksi.mff.cuni.cz

Abstrakt: V oblasti bezdrôtových sietí existuje a je používané množstvo alternatívnych smerovacích protokolov oproti protokolom používaným v statických sieťach. Jedným z nich je aj protokol Ad Hoc On-Demand Distance Vector (AODV), ktorý sa používa v mobilných ad-hoc sieťach, to je v sieťach bez pevnej komunikačnej infraštruktúry, kde sú všetky uzly medzi sebou navzájom rovnocenné a podieľajú sa na preposielaní správ. AODV protokol používa ako kritérium pri výbere trás počet uzlov na trase, čo v bezdrôtových sieťach nie je vždy optimálne kritérium. Táto práca vznikla za účelom návrhu a otestovania prevádzky tohto smerovacieho protokolu s využitím iných ohodnocovacích kritérií ako je dĺžka trasy. Súčasťou práce je aj testovacia implementácia navrhnutých riešení a simulácia ich prevádzky vo vybranom simulačnom prostredí. V závere práce sú popísané dosiahnuté výsledky a ich pozitívne alebo negatívne vlastnosti.

Klíčová slova: Ad Hoc On-Demand Distance Vector smerovací protokol, bezdrôtové ad-hoc siete, smerovacie protokoly, metriky smerovacích protokolov

Title: Implementation of alternative metric in AODV protocol
Author: Matej Dajčár
Department: Department of Software Engineering
Supervisor: Mgr. Miroslav Novotný
Supervisor's e-mail address: novotny@ksi.mff.cuni.cz

Abstract: There is a lot of alternative routing protocols used in wireless communications. One of these protocols is Ad Hoc On-Demand Distance Vector routing protocol (AODV). This protocol is used in the mobile ad-hoc networks which are self-configuring networks consisting of the independent mobile devices where each one of these devices acts as a router and forwards traffic from other devices. AODV protocol uses hop count as a routing metric, but in the many cases this metric is not optimal in the wireless networks. The goal of this thesis is to propose the alternative criteria which can be used to select best routes. An integral part of this thesis is the experimental implementations of suggested metrics which will be simulated and evaluated in the selected simulation tool. The conclusion of the thesis analyses results obtained from the simulations of the individual suggested versions.

Keywords: Ad Hoc On-Demand Distance Vector Routing, ad-hoc networks, routing protocols, routing metrics

Kapitola 1

Úvod

Potreba ľudí komunikovať medzi sebou spojená s rapídnym technologickým vývojom v posledných desaťročiach so sebou priniesli nové komunikačné možnosti. V počiatkoch počítačových sietí bola komunikácia medzi jednotlivými sieťovými zariadeniami alebo pripojenie do internetu možné len na miestach, kde existovala a bola dostupná pevná komunikačná infraštruktúra. S postupným technickým vývojom sa do popredia a obľuby dostávali zariadenia, ktoré umožňujú komunikáciu medzi užívateľmi a prístup k informáciám aj počas mobility užívateľa. Týmto zaniká obmedzenie, keď sa počas pripojenia musí užívateľ nachádzať na jedom mieste alebo v jeho blízkom okolí v dosahu pevnej sieťovej infraštruktúry. V rámci tohto vývoja vzniklo množstvo komunikačných technológií, štandardov a protokolov, ktoré sú určené pre podporu bezdrôtového prenosu dát. Bezdrôtové siete poskytujú oproti klasickým pevným káblovým sieťam viacero výhod, ale majú aj svoje nevýhody. K najväčším výhodám patrí možnosť pohybu používateľa, možnosť vytvorenia tohto druhu siete skoro na ľubovoľnom mieste, jednoduchosť a rýchlosť pripojenia sa k takejto sieti a absencia finančných nákladov a problémov spojených s inštaláciou a údržbou kabeláže. Z nevýhod sú to u väčšiny technológií problémy spojené s použitím odlišného prenosového média, z čoho vyplývajú obmedzenia rýchlosti, spoľahlivosti a kvality bezdrôtového prenosu, ako aj možnosti zdieľania prenosového média. Nevýhodou bezdrôtových sietí je aj nižšia úroveň bezpečnosti ako tá, ktorú je možné dosiahnuť v sieti založenej na pevnej infraštruktúre. Na základe týchto rozdielov nie je možné v bezdrôtových sieťach efektívne používať všetky mechanizmy a protokoly, ktoré boli vyvinuté pre prostredie statických sietí. Bolo potrebné navrhnúť nové architektúry a protokoly, ktoré by boli čo najlepšie prispôsobené odlišným vlastnostiam bezdrôtového prenosu, využívali jeho prednosti a v čo najvyššej miere boli schopné eliminovať nevýhody tohto prenosu. Jedným z typov bezdrôtových sietí je sieť typu Ad-hoc, ktorou sa zaoberá aj táto práca. Kvôli špecifickým vlastnostiam tohto typu sietí boli pre túto sieť vytvorené osobitné smerovacie protokoly, jednému vybranému a jeho modifikácii sa budem v tejto práci venovať.

Cieľom tejto práce bolo detailné oboznámenie sa so smerovacím protokolom Ad Hoc On-Demand Distance Vector Routing (AODV) a návrh alternatívnych metrík, na základe ktorých smerovací protokol ohodnocuje a vyberá cesty v sieti, po ktorých následne posiela sieťové prenosy. Smerovací protokol AODV používa ako metriku počet uzlov nachádzajúcich sa na ohodnocovanej trase. Motívom práce bolo nahradenie tejto základnej metriky metrikami založenými na odlišnom kritériu ako je dĺžka trasy a pozorovanie aký vplyv majú jednotlivé metriky na prevádzku v sieti. Súčasťou práce by malo byť aj porovnanie výkonu siete pri použití alternatívnej metriky oproti výkonu siete s použitou pôvodnou metrikou.

Na začiatku tejto práce je v úvodnej kapitole uvedený stručný prehľad a popis bezdrôtových sietí spolu s klasifikáciou sietí typu ad-hoc, popisom ich charakteristických vlastností a prehľadom jednotlivých typov ad-hoc sietí. Kapitola 2.3 obsahuje rozdelenie smerovacích protokolov používaných v ad-hoc sieťach.

V tretej kapitole je uvedený detailný popis funkčnosti smerovacieho protokolu Ad Hoc On-Demand Distance Vector Routing a popis typov jednotlivých správ, ktoré k svojmu fungovania protokol využíva.

Štvrtá kapitola obsahuje popis metrík, ktoré boli v rámci tejto práci vybrané alebo navrhnuté a následne implementované. Pri každej metrike je uvedená idea, na základe ktorej bola metrika navrhnutá a následne zmeny, ktoré bolo potrebné pri implementácii metriky vykonať.

Piata kapitola obsahuje prehľad simulačných nástrojov a prostredí, ktoré boli vyvinuté pre simuláciu sieťového prostredia a komunikácie v ňom. V tejto kapitole sú uvedené aj dôvody pre výber nástroja, ktorý som použil pre otestovanie navrhnutých metrík. Kapitola obsahuje aj zoznam a popis existujúcich implementácií AODV protokolu, z ktorých bola jedna použitá ako základ, do ktorého som implementoval potrebné zmeny. Koniec piatej kapitoly obsahuje popis simulačných scenárov, v ktorých boli prevádzané testy ako aj popis sledovaných kritérií, na základe ktorých som vyhodnocoval výkon jednotlivých modifikácií protokolu.

V šiestej kapitole sú ku každej modifikovanej verzii zobrazené výsledky vybraných simulácií obsahujúce doplňujúci popis. V poslednej kapitole je zhrnutie celej práce a celkové zhodnotenie dosiahnutých výsledkov ako aj námety na možné nadviazanie tejto práce.

Kapitola 2

Ad-hoc siete a smerovanie v ad-hoc sieťach

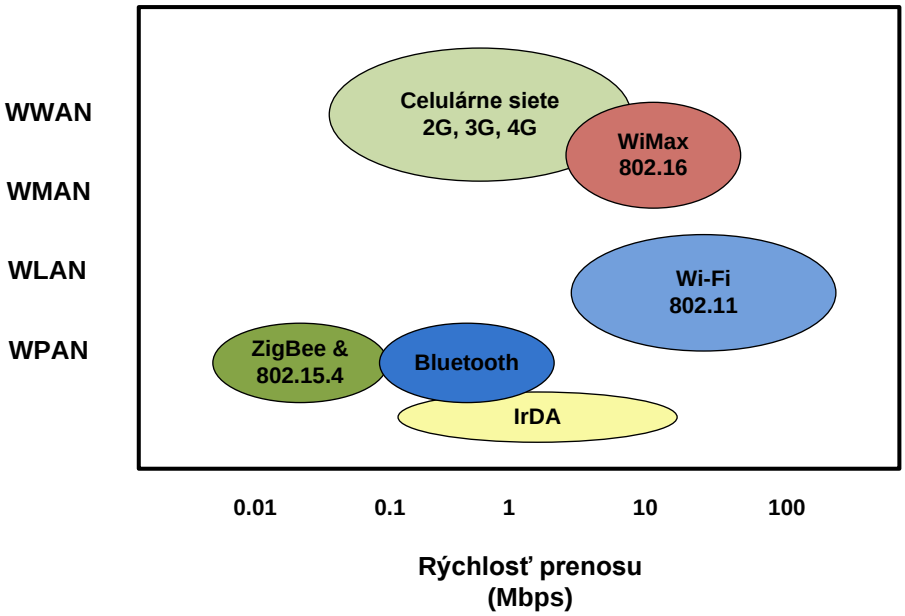
Bezdrôtová ad-hoc sieť je decentralizovaná sieť vytvorená z uzlov používajúcich ako médium pre prenos komunikácie vzduch. Ad-hoc sieť obvykle nemá dopredu pripravenú žiadnu infraštruktúru a vzniká za určitým konkrétnym účelom. V sieti tohto typu neexistuje žiaden centrálny prvok, ktorý riadi alebo organizuje chod siete, všetky uzly v sieti sú si navzájom rovné. Každý z uzlov siete zastáva funkciu koncového uzla a okrem toho môže zastávať aj funkciu smerovača (routra), ktorý sa podieľa na prenose komunikácie medzi uzlami, ktoré nie sú v priamom dosahu svojho vysielania. Ak by sa uzly na takejto komunikácii medzi sebou aktívne nepodieľali - v sieti by bola možná komunikácia len medzi dvojicami prvkov, u ktorých vysielanie dosiahne k protiľahlému susednému uzlu. Susednými uzlami budem nazývať uzly, ktoré sú v priamom dosahu svojho vysielania a k prenosu komunikácie medzi sebou nepotrebujú žiadneho prostredníka. Jednotlivé uzly sa môžu väčšinou v ad-hoc sieti pohybovať, takže topológia siete sa môže s plynúcim časom meniť.

Z hľadiska rozdelenia bezdrôtových sietí ad-hoc siete najčastejšie patria do kategórie Wireless Local Area Networks (WLAN), prípadne podľa použitého zariadenia môžu v niektorých prípadoch patriť do skupín Wireless Personal Area Networks (WPAN) alebo Wireless Metropolitan Area Networks (WMAN).

Siete typu WPAN sa používajú na komunikáciu na krátke vzdialenosti, zariadeniami využívanými pri tomto type sietí sú napr. technológie založené na štandardoch Bluetooth (IEEE 802.15.1), infračervený port (IrDA) alebo ZigBee. Charakteristickými vlastnosťami je nízka prenosová rýchlosť v jednotkách Mbit/s a vysielací dosah takisto v jednotkách metrov.

Siete typu WLAN sú siete stredného dosahu, obvykle z rozsahu desiatok až stoviek metrov, založené na skupine štandardov IEEE 802.11. Toto označenie zahŕňa množstvo štandardov, z ktorých hlavné a v súčasnosti najčastejšie používané sú 802.11a, 802.11b, 802.11g a 802.11n. Prenosové rýchlosti v týchto sieťach sa pohybujú približne v rozsahu 10 až 50 Mbit/s, pri štandarde 802.11n sa použiteľné rýchlosti môžu dostať až na hodnotu okolo 150 Mbit/s.

Siete typu WMAN sa používajú na komunikáciu na väčšie vzdialenosti, môžu byť použité



Obr. 2.1: Klasifikácia bezdrôtových sietí

napríklad na prepojenie viacerých WLAN sietí. Predstaviteľom tejto skupiny je napríklad technológia založená na štandarde WiMax (IEEE 802.16). V praxi je typicky signálom jedného vysielača pokryté územie v rozsahu 5-15 km, kde sa rýchlosti pre koncových užívateľov pohybujú v jednotkách Mbit/s.

2.1 Charakteristické vlastnosti ad-hoc sietí

Ad-hoc sieť má charakteristické vlastnosti, ktorými sa líši od klasických centralizovaných sietí alebo sietí s pevnou infraštruktúrou. V nasledujúcich bodoch sú uvedené hlavné z týchto odlišujúcich charakteristík.

Mobilita
U uzlov ad-hoc siete sa predpokladá, že sa v prostredí môžu pohybovať a môžu do siete ľubovoľne vstupovať alebo vystupovať. Pohybom uzlov môžu vznikať v sieti nové spojenia, podľa toho, ktoré uzly sú aktuálne v spoločnom dosahu svojho vysielania, taktiež môžu na existujúcich trasách spoje náhodne zanikať. V prípade zániku alebo prerušenia spojenia je obvykle potrebné prerušené spojenie obnoviť alebo nájsť alternatívnu trasu medzi koncovými uzlami, po ktorej by bol možný prenos prerušenej komunikácie. Uzly sa môžu v sieti pohybovať buď náhodne alebo po určitých typických trasách, napríklad ako sa pohybujú ľudia v meste alebo autá po cestách. Charakter pohybu môže byť individuálny pre každý samostatný uzol alebo spoločný pre viacero uzlov, napríklad keď je uzol súčasťou skupiny, kde majú všetky uzly rovnaké správanie - príkladom takéhoto pohybu môže byť skupina osôb na spoločnom výlete.

Organizácia a konfigurácia uzlov v sieti
Ad-hoc sieť musí obsahovať samo-organizačný mechanizmus, ktorý zaistí začlenenie nových prvkov do siete alebo zhodnutie sa na spoločných parametroch jednotlivých uzlov potrebných pre funkčnosť siete. Keďže v sieti neexistuje žiaden centrálny riadiaci mechanizmus, ktorý by riadil nastavenie konfiguračných parametrov siete, musia sa uzly na týchto konfiguračných parametroch dohodnúť medzi sebou sami. Na to je potrebné definovať spoločné rozhranie, pomocou ktorého si môžu uzly tieto konfiguračné parametre predávať. Príkladom parametrov potrebných pre fungovanie siete môže byť systém adresácie uzlov v sieti, typ použitého smerovacieho protokolu alebo parametre potrebné pre optimálne fungovanie vybraných protokolov, ako sú napríklad interval medzi aktualizáciami smerovacích tabuliek alebo priemer siete.

Energetický zdroj uzlov v ad-hoc sieti
U mobilných uzlov sa predpokladá obmedzený zdroj energie. Z toho vyplýva časové obmedzenie, ako dlho môže daný uzol v sieti fungovať a potreba minimalizovať spotrebu energie za účelom maximalizácie doby, po ktorú je uzol schopný komunikovať. Ad-hoc sieť je ako celok schopná fungovať len tak dlho, pokiaľ obsahuje uzly, ktoré sú schopné podieľať sa na prenose potrebnej komunikácie.

Bezpečnosť siete
Ad-hoc sieť je podstatne náchylnejšia k bezpečnostným hrozbám ako centralizovaná sieť alebo sieť s pevnou infraštruktúrou. Nižšia bezpečnosť vyplýva z použitého média, kde komunikáciu, ktorú uzol vysiela, sú schopné zachytiť všetky uzly nachádzajúce sa v dosahu vysielania. Ďalšie riziko vyplýva z princípu, akým funguje prenos dát v ad-hoc sieti. Keďže na prenose komunikácie sa podieľajú rôzne uzly na trase od zdrojového uzlu k cieľovému uzlu, nie je zložité narušiť alebo ohroziť komunikáciu medzi koncovými uzlami (man-in-the-middle attack, denial of service). Potencionálne škodlivý uzol nachádzajúci sa na trase medzi zdrojovým a cieľovým uzlom má možnosť všetky vybrané správy zahadzovať, prípadne posielať iným smerom ako sa nachádza správny cieľový adresát. Rušenie komunikácie sa môže diať buď úmyselne, kde uzol aktívne ruší komunikáciu medzi danými uzlami alebo neúmyselne. Prípad neúmyselného škodenia môže byť napríklad zahadzovanie paketov pri stave vybitej batérie, ak je daný uzol nastavený, aby sa pri nízkej úrovni energie nepodieľal na komunikácii, ktorú nevyžaduje primárne pre svoje účely.

Heterogenita komunikujúcich uzlov
Jednotlivé uzly ad-hoc siete môžu byť značne odlišné z pohľadu technických parametrov, napríklad v prenosovej kapacite uzla, výkonu vysielacieho zariadenia, maximálnej rýchlosti, akou je uzol schopný pohybovať sa alebo v kapacite batérie. V niektorých prípadoch môže táto rôznorodosť ovplyvňovať funkčnosť siete alebo spôsobovať problémy, na riešenie ktorých musí sieť obsahovať adekvátne prostriedky.

Kvalita služieb
Rôzne uzly v sieti môžu požadovať garanciu kvality služieb na rôznych úrovniach v závislosti na type komunikácie alebo charaktere dát, ktoré prenášajú. Vzhľadom k mobilite uzlov

v sieti, premenlivým podmienkam okolitého prostredia, ako je zdieľanie média pre prenos signálu, meniaca sa kvalita signálu alebo vzhľadom k rôznorodosti uzlov v sieti, môže byť zabezpečenie kvality služieb na úrovni požadovanej jednotlivými uzlami väčším problémom ako u sietí s pevnou infraštruktúrou.

Škálovateľnosť
Ad-hoc sieť musí byť schopná spĺňať svoju funkciu aj pri rapídnom náraste celkového počtu uzlov v sieti alebo náraste počtu uzlov, ktoré chcú komunikovať zároveň. Réžia na zaistenie funkčnosti prenosu v sieti sa musí pohybovať na minimálnych možných úrovniach, aby nedochádzalo k zbytočnému obsadzovaniu média alebo znižovaniu kapacity siete správami potrebnými pre zaistenie správneho smerovania.

Vzhľadom k uvedeným charakteristikám a z nich vyplývajúcim obmedzeniam je potrebné pri návrhu algoritmov v ad-hoc sieťach myslieť na obmedzenie týchto možných nedostatkov. Týka sa to hlavne návrhu a úpravy algoritmov používaných na linkovej a sieťovej vrstve OSI modelu

2.2 Typy ad-hoc sietí

Existuje viac typov ad-hoc sietí deliacich sa podľa účelu pre ktorý sieť slúži, alebo podľa typov zariadení, z ktorých sa sieť skladá. Nasledujúce uvedené sú hlavné z týchto typov.

2.2.1 Mobile Ad-hoc Networks (MANET)

MANET je typ siete odpovedajúci popisu z úvodu 2. kapitoly. Takáto sieť sa skladá z nezávislých bezdrôtovo komunikujúcich zariadení, ktoré majú schopnosť mobility. Sieť je schopná fungovať bez akejkoľvek pevnej infraštruktúry, je založená len na bezdrôtovej komunikácii uzlov medzi sebou. Rýchlosť, akou sa zariadenia môžu pohybovať, môže byť v rozsahu od nuly pri statických uzloch až po desiatky kilometrov za hodinu v prípade, že uzlami v sieti sú napríklad dopravné prostriedky. Veľkosť siete sa môže pohybovať od siete s jednotkami uzlov až po siete obsahujúce tisícky uzlov. Siete typu MANET vznikli z dôvodu potreby umožnenia komunikácie jednotiek pri vojenských cvičeniach alebo operáciách, napríklad na cudzom území, kde je potrebné zabezpečiť komunikáciu mobilných jednotiek medzi sebou. Ďalším možným použitím je zabezpečenie komunikácie pri záchranných operáciách, ktorých sa zúčastňuje viac jednotiek. Možným príkladom použitia je aj použitie v komerčnej sfére, napríklad na vytvorenie komunikačnej siete pri obchodných rokovaniach alebo konferenciách na miestach, kde neexistuje žiadna sieťová infraštruktúra a kde sa neoplatí pre dočasné použitie infraštruktúru inštalovať.

Špeciálnym typom skupiny MANET je ad-hoc sieť tvorená dopravnými prostriedkami cestnej premávky. Táto sieť sa nazýva *Vehicular ad-hoc network (VANET)* alebo verzia *InVANET (Intelligent VANET*). Tento typ siete je v posledných rokoch predmetom mnohých výskumov. Podľa návrhov by automobily obsahovali bezdrôtové vysielače, pomocou ktorých by sa stali súčasťou komunikačnej siete, v ktorej by sa medzi sebou mohli nezávisle informovať o situácii v cestnej premávke, haváriách, zápchach alebo dopravných obmedzeniach.

2.2.2 Wireless Mesh Networks (WMN)

Typ siete podobný typu MANET, ale sieť môže okrem mobilných zariadení obsahovať aj uzly pripojené do fixnej infraštruktúry. Mesh sieť obsahuje 2 typy uzlov a to klientské uzly (mesh clients) a smerovacie uzly (mesh routers). Klientské uzly sú väčšinou prenosné počítače, PDA, mobilné telefóny a zariadenia podobného typu, pričom smerovacie stanice môžu byť výkonnejšie počítače alebo prístupové body používané v WLAN sieťach. Smerovacie uzly vytvárajú v bezdrôtovej sieti hlavné spoje, po ktorých sa prenáša komunikácia (obdoba tzv. backbone siete) a môžu slúžiť aj ako komunikačné brány a zaisťovať konektivitu do internetu alebo častí siete dostupných len cez pevnú infraštruktúru. Klienti majú možnosť komunikovať priamo medzi sebou alebo pomocou smerovacích uzlov. Jeden z rozdielov medzi klientskými a smerovacími stanicami je, že smerovacie stanice nemusia podliehať obmedzeniam, ktorým podliehajú klientské stanice ako sú napríklad obmedzený zdroj energie, obmedzená prenosová kapacita alebo limitovaný výpočtový a prenosový výkon. Príkladom WMN siete môže byť sieť vytvorená za účelom zdieľania prístupu do internetu.
Do skupiny WMN sa dá zaradiť aj varianta siete VANET /InVANET a to v prípade, ak cestná sieť obsahuje aj statické komunikačné uzly určené na komunikáciu s automobilmi. Tieto uzly môžu zaisťovať prenos informácii napr. z dopravnej centrály alebo z internetu.

2.2.3 Wireless Sensor Networks (WSN)

Tento typ sietí bol navrhnutý pre monitorovanie a zber dát pomocou uzlov obsahujúcich senzory. Senzorová sieť sa väčšinou skladá z veľkého počtu uzlov vytvárajúcich senzorové pole, každý uzol môže obsahovať jeden alebo viac senzorov. Každý takýto autonómny uzol môže monitorovať okolie, snímať a spracovávať z neho informácie a tie následne posielať ďalej buď v originálnej alebo spracovanej forme. Dáta, ktoré uzly pomocou senzorov z okolia nasnímajú, posielajú zvolenému centrálnemu uzly, ktorý ich sprístupňuje užívateľom. V niektorých prípadoch môže byť sieť členená do hierarchií, kde zberné uzly propagujú informáciu hierarchicky vyššie postaveným uzlom a tie propagujú dáta zase smerom vyššie. Senzorové siete sú často využívané v priemysle, zdravotníctve, na vojenské účely alebo napríklad na monitorovanie prostredia v prírode.

2.3 Smerovacie protokoly v ad-hoc sieťach

Smerovanie v ad-hoc sieťach kladie na smerovacie algoritmy nové požiadavky, ktoré v sieťach s pevnou infraštruktúrou nebolo potrebné brať do úvahy. Algoritmy, ktoré boli pôvodné vyvinuté pre fixné siete nemuseli brať do úvahy mobilitu prvkov, dynamicky sa meniaci tvar siete, častý vznik a zánik trás alebo variabilitu podmienok prostredia, v ktorom sa komunikácia uskutočňuje. Všetky tieto vlastnosti sa v ad-hoc sieťach vyskytujú a treba na ne brať pri návrhu smerovacích protokolov ohľad.
Smerovací algoritmus by pre optimálne fungovanie mal spĺňať nasledujúce vlastnosti:

Optimalita - vybrané cesty by mali byť z pohľadu zvolenej metriky najoptimálnejšie vzhľadom k ostatným možným cestám.

Efektivita algoritmu, nízka réžia na zaistenie smerovania - algoritmus by nemal zbytočne zaťažovať procesor a sieťové zdroje nadmerným počtom operácií a vyslaných kontrolných správ. Toto kritérium má u bezdrôtových sietí väčšiu váhu, vzhľadom k tomu že uzly komunikujú pomocou zdieľaného média a majú obmedzený zdroj energie.

Robustnosť, stabilita a flexibilita - algoritmus by mal fungovať aj za nezvyčajných alebo nepredvídateľných okolností, ako napríklad nefunkčnosť niektorých uzlov v sieti, vysoké vyťaženie uzlov alebo rýchly nárast alebo úbytok počtu uzlov nachádzajúcich sa v sieti.

Rýchla konvergencia - konvergencia znamená, že všetky uzly majú optimálne a správne smerovacie informácie. Ak majú uzly nesprávne smerovacie informácie, algoritmus nemusí fungovať správne, dôsledkom čoho môžu vznikať napríklad smerovacie cykly alebo miesta v sieti, kde sú zahadzované pakety.

V ad-hoc sieťach používané smerovacie protokoly sa delia do troch hlavných kategórií, a to na proaktívne, reaktívne a hybridné smerovacie protokoly. Toto delenie je určené podľa spôsobu, akým smerovacie protokoly jednotlivých kategórií získavajú a udržujú smerovacie informácie.

2.3.1 Proaktívne smerovacie protokoly

Proaktívne smerovacie protokoly udržiavajú v smerovacej tabuľke každého uzla aktuálne informácie o dostupnosti všetkých ostatných uzlov (alebo sa aspoň snažia o to, aby boli uložené informácie aktuálne). Uzly si medzi sebou musia propagovať zmeny, ktoré sa vyskytli v sieti, aby sa aj ostatné uzly dozvedeli o týchto zmenách a mohli si aktualizovať smerovacie informácie. Nevýhodou týchto smerovacích protokolov je, že uzly musia udržiavať aktuálne informácie o trase aj k tým uzlom, ku ktorým nevysielajú a v danom okamihu teda dané smerovacie informácie nepotrebujú. Z tohto vyplývajú vyššie režijné náklady, ktoré sú spojené s propagáciou všetkých aktualizácií skrz celú sieť. Taktiež konvergencia smerovacích tabuliek všetkých uzlov do stavu, v akom sa aktuálne sieť nachádza, môže zabrať dlhší čas. Výhodou proaktívnych smerovacích protokolov je, že všetky uzly majú väčšinu času aktuálnu smerovaciu informáciu k ostatným uzlom v sieti. V prípade požiadavku na komunikáciu nevzniká žiadne zdržanie zapríčinené hľadaním použiteľnej trasy, uzol môže okamžite začať vysielať správy. Použitie tohto typu protokolu sa neodporúča v sieťach s vysokou frekvenciou topologických zmien, kde môže byť značná časť prenosovej kapacity zahltená propagáciou smerovacích informácii, ktorých spracovanie znižuje výkon a energiu jednotlivých uzlov siete. K smerovacím protokolom tejto skupiny používaným v prostredí MANET sietí patria napríklad *Destination-Sequenced Distance-Vector protocol (DSDV)*[32] alebo *Optimized Link-State Routing protocol (OLSR)*[17]. Proaktívne smerovacie protokoly zvyknú byť označované aj ako *Table driven routing protocols.*

2.3.2 Reaktívne smerovacie protokoly

Reaktívne smerovacie protokoly na rozdiel od proaktívnych protokolov neudržujú stále smerovacie informácie k ostatným uzlom v sieti. Tento typ býva označovaný aj ako *Source initiated routing protocol.* Reaktívne protokoly začínajú hľadať trasu až pri vzniku požiadavku na komunikáciu s cieľovým uzlom. Týmto mechanizmom protokol eliminuje nadbytočnú ré-

žiu potrebnú k propagácii zmien v sieti a udržovaniu aktuálnych smerovacích informácií. Zmena topológie v časti siete, ktorú uzol nepoužíva, ho nemusí zaujímať, preto táto zmena nie je premietnutá do jeho smerovacích tabuliek. Nevýhodou je, že začiatok komunikácie s novým uzlom môže sprevádzať dlhší čas potrebný k nájdeniu použiteľnej trasy, po ktorej by komunikácia mohla prebiehať. Kvôli tejto vlastnosti nie je použitie reaktívneho protokolu vhodné v sieti, kde sa často prenášajú real-time dáta. Nevýhodou môže byť časté zaplavovanie siete požiadavkami na zistenie trasy. Toto zaplavovanie môže byť obmedzené použitím vhodných techník pre obmedzenie rozsahu, v ktorom sú jednotlivé požiadavky vysielané. Reaktívne smerovacie protokoly viac šetria prenosovou kapacitou média ako aj čerpaním energie jednotlivých uzlov. Predstaviteľmi tejto kategórie protokolov sú napríklad *Dynamic Source Routing*[25], *Temporarily Ordered Routing Algorithm*[39] alebo *Ad Hoc On-Demand Distance Vector Routing*[33], ktorý je podrobnejšie popísaný v kapitole č. 3.

2.3.3 Hybridné smerovacie protokoly

Pri vzniku hybridných smerovacích protokolov stála snaha skombinovať výhody reaktívnych a proaktívnych smerovacích protokolov. Protokoly tohto typu využívajú obidva spôsoby na správu smerovacích tabuliek. V častiach siete, kde sa nepredpokladá častá zmena topológie alebo v častiach, ktoré sú pre uzol zaujímavejšie ako ostatné, môžu uzly udržovať smerovacie informácie proaktívnym spôsobom. Smerovacie informácie vzťahujúce sa k časti siete, kde sa predpokladá vyššia dynamika, môžu uzly získavať a udržovať reaktívnym spôsobom. Predstaviteľmi tohto typu smerovania sú napríklad protokoly *Zone Routing Protocol*[41] alebo *Fisheye State Routing protocol*[31].

Cieľom jednotlivých smerovacích protokolov je zaistenie požadovanej konektivity, aby uzly v sieti vedeli určiť smer alebo trasu, ktorými majú komunikovať s vybranými uzlami. Pri zaisťovaní tejto konektivity môžu rôzne protokoly uprednostňovať rôzne kritéria závislé na type siete alebo prenášanej komunikácie. Väčšinou sú požadovanými kritériami čo najvyššia úspešnosť doručenia vyslaných paketov, čo najvyššia priepustnosť siete, nízke hodnoty zdržania vyslaných správ a rozptyl hodnôt tohto zdržania a to všetko za použitia nízkych režijných nákladov na smerovanie.

Kapitola 3

Smerovací protokol Ad-hoc On Demand Distance Vector

Smerovací protokol Ad-hoc On Demand Distance Vector patrí do skupiny reaktívnych smerovacích protokolov typu hop-by-hop routing. Úlohou AODV je zabezpečiť aktuálne a optimálne smerovacie informácie uzlu, ktorý chce v danom momente vysielať. AODV protokol je definovaný v RFC 3561 [7], na základe návrhu vypracovaného p. C. Perkinsom. Na vyhľadávanie najkratších ciest je v AODV použitý Bellman-Fordov algoritmus. V tomto protokole sú najkratšie trasy vyberané podľa počtu uzlov, nachádzajúcich sa na danej trase. AODV je navrhnutý pre použitie v sieti obsahujúcej mobilné uzly, aj preto obsahuje mechanizmy pre rýchlu opravu prerušenej trasy alebo vyhľadanie trasy novej. Uzly si uchovávajú aktuálne informácie len o tých trasách, ktoré sú aktuálne používané. Smerovacie informácie vzťahujúce sa k uzlom, cez ktoré po určitý čas nie sú prenášané žiadne dáta, sú po uplynutí stanovenej doby zo smerovacej tabuľky odstránené. Na zabezpečenie aktuálnosti smerovacích informácií a na zamedzenie tvorby smerovacích cyklov algoritmus používa mechanizmus sekvenčných čísel. Každý uzol má vlastné sekvenčné číslo, ktorého hodnotu za špecifických okolností inkrementuje. V smerovacej tabuľke je pri každom zázname zaznamenané posledné známe číslo, ktoré uzol, ku ktorému sa viaže záznam, má alebo mal. Sekvenčné čísla záznamov smerovacej tabuľky uzol aktualizuje podľa prijatých AODV správ. Čím vyššie je sekvenčné číslo, tým novšia je daná informácia. Pri príjme ľubovoľnej AODV správy, ktorá sa týka niektorého záznamu v smerovacej tabuľke, uzol aktualizuje sekvenčné číslo daného záznamu na aktuálnejšiu hodnotu z hodnoty, ktorú má zaznamenanú v smerovacej tabuľke a z hodnoty obsiahnutej v prijatej správe. Aktuálnejšia informácia je tá, ktorá má vyššie sekvenčné číslo.

Uvediem terminológiu, ktorú budem v nasledujúcom popise používať. Typy správ uvedené v nasledujúcich odstavcoch budú popísané v ďalších odsekoch.

- zdroj/zdrojový uzol - uzol, ktorý hľadá trasú k ľubovoľnému uzlu. Tento uzol generuje správu typu *Route Request* a vysiela ju do siete. V Route Request správe je uvedený ako pôvodca tejto správy, t.j. (*Route Request Originator*)
- cieľ/cieľový uzol - uzol, do ktorého je hľadaná cesta. V Route Request je uvedený ako cieľový uzol, t.j. (*Route Request Destination*)

- Vzdialenosťou alebo dĺžkou určitej trasy medzi uzlami A a B budem nazývať počet uzlov, ktoré táto trasa obsahuje - sú to uzly, ktoré sa podieľajú na prenose komunikácie z uzlu A do uzlu B
- medziuzol - uzol nachádzajúci sa na vytvorenej trase medzi zdrojovým a cieľovým uzlom

3.1 Popis algoritmu

Smerovací protokol používa pre svoj chod smerovaciu tabuľku, v ktorej si udržuje smerovacie záznamy s informáciami o ostatných uzloch siete. Každý záznam v smerovacej tabuľke obsahuje adresu cieľového uzla, ku ktorému sa daný záznam vzťahuje, vzdialenosť k tomuto uzlu a adresu najbližšieho uzla, kam sa majú preposielať pakety adresované cieľovému uzlu (tento uzol sa označuje ako next hop). Záznam ďalej obsahuje informáciu o čase, ako dlho je trasa k cieľovému uzlu považovaná za aktívnu. Pri každom poslaní paketu po danej trase sa táto doba aktualizuje na aktuálny čas zvýšený o definovanú hodnotu. V smerovacom zázname sú taktiež uložené adresy susedných uzlov, ktoré používajú tento smerovací záznam na prenos paketov. Týmto uzlom sa v prípade prerušenia cesty posiela správa o chybe v spojení. Tieto uzly budem označovať ako predchodcov smerovacieho záznamu. Uzly, ktoré používajú AODV protokol, môžu mať viac sieťových rozhraní s rozličnými adresami, ktorými komunikujú v sieti - preto by mala byť pre tento prípad v smerovacom zázname uložená aj identifikácia rozhrania, pomocou ktorého uzol získal informácie viažuce sa k smerovaciemu záznamu.

K vyhľadávaniu a spravovaniu smerovacích záznamov používa AODV protokol smerovacie správy. AODV definuje 3 druhy používaných smerovacích správ

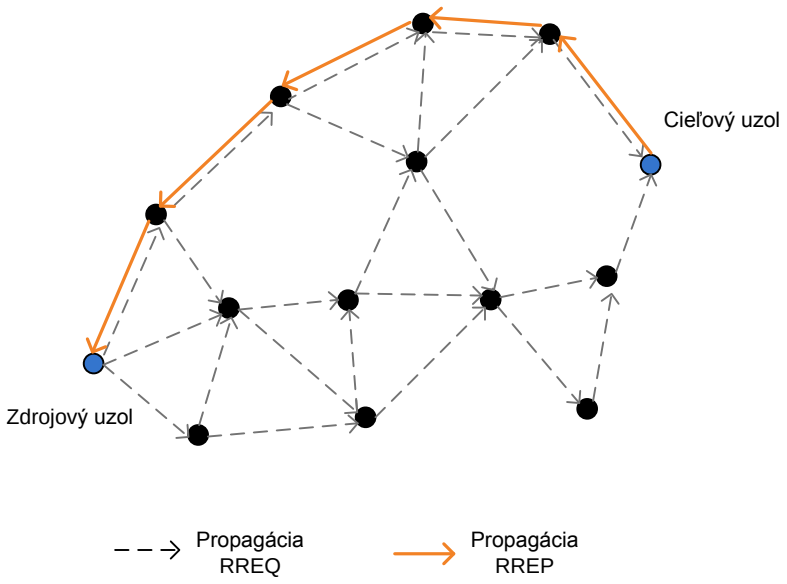
- Route Request (RREQ), ktoré sú použité k vyhľadávaniu ciest
- Route Reply (RREP), ktoré sú použité ako odpoveď na správy Route Request
- Route Error (RERR), ktoré sa používajú na oznámenie o prerušení existujúcej cesty

Kvôli špecifickému použitiu môžu byť ako dodatočný typ považované aj HELLO správy, aj keď samotné správy sú typu Route Reply.

3.1.1 Generovanie Route Request

Pri spracovaní paketu odchádzajúceho zo sieťovej vrstvy musí uzol zistiť, kam má byť paket odoslaný. Paket môže byť adresovaný a odoslaný všetkým uzlom (tzv. broadcast), alebo môže byť adresovaný len jednému uzlu (tzv. unicast), prípadne vybranej skupine uzlov (tzv. multicast). V prípade unicastu je potrebné zistiť, ktorému ďalšiemu uzlu má byť daný paket poslaný. Ak sieťová vrstva dostane odoslať paket pre príjemcu, u ktorého nevie kam má príslušné pakety posielať, musí zistiť adresu nasledujúceho uzla na tejto ceste. To sa v AODV deje pomocou vygenerovania a vyslania správy typu Route Request. RREQ je generovaná v prípade, že uzol nemá žiadnu smerovaciu informáciu o cieľovom uzle alebo má informáciu, ktorá je už neplatná - to môže nastať vtedy, ak uzol mal aktuálnu informáciu ale daný záznam

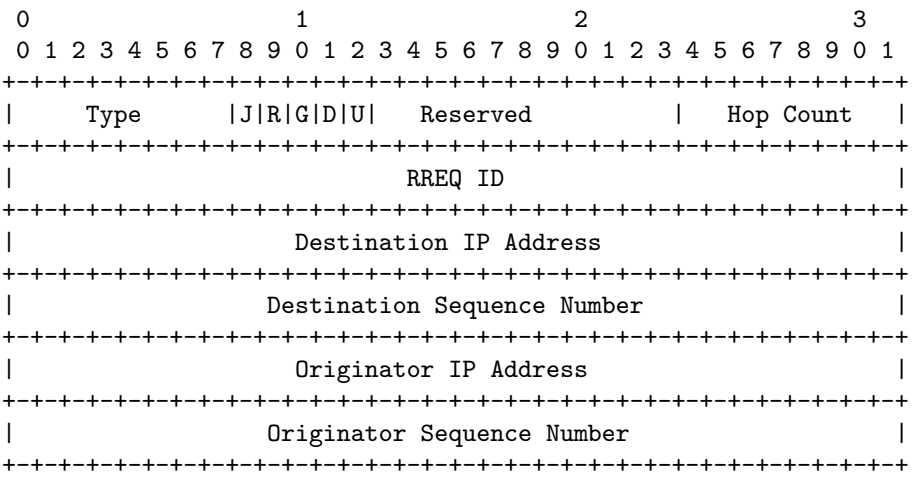
nebol určitý čas používaný ani aktualizovaný, čím sa dostal do neplatného - expirovaného stavu. V AODV je tento stav označovaný ako *invalid*.



Obr. 3.1: Propagácia Route Request a Route Reply správ

V každej RREQ správe je uvedená adresa uzla-pôvodcu, ktorý danú RREQ vygeneroval, sekvenčné číslo tohto pôvodcu, adresa cieľového uzla, do ktorého sa hľadá cesta a posledné známe sekvenčné číslo, ktoré mal pôvodca zaznamenané pri cieľovom uzle. Ak uzol, ktorý generuje RERQ, nemá doteraz žiadnu informáciu o sekvenčnom čísle cieľového uzla, nastaví v RREQ príznak, že je sekvenčné číslo cieľa neznáme. Okrem týchto informácii RREQ ešte obsahuje hodnotu Hopcount, v ktorej je uvedené, koľkými uzlami bola RREQ preposlaná, ďalej informácie používané pri multicastovom vysielaní a ďalšie príznaky uvedené v schéme 3.2.
Predtým, ako uzol vygeneruje RREQ, zvýši svoje sekvenčné číslo a takisto aj sekvenčné číslo cieľového uzlu, ak má pri tomto cieľovom uzle nejaké číslo zaznamenané. Zvýšenie sekvenčných čísel zaručí, že pri spracovaní vyslanej RREQ ostatnými uzlami nevzniknú v sieti smerovacie cykly a odpoveď prijatá na základe vygenerovanej RREQ bude obsahovať aktuálne informácie o cieľovom uzle. Hodnota *Hopcount* je pri vygenerovaní RREQ nastavená na 0. Pred poslaním vygenerovanej RREQ do siete si uzol zapamätá identifikáciu tejto správy. Ako identifikátor RREQ je použitá dvojica [adresa pôvodcu; RREQ ID]. RREQ ID je sekvenčné alebo poradové číslo RREQ, ktoré má každý uzol vlastné a inkrementuje si ho pri každom vytvorení RREQ. Uzol si pamätá všetky RREQ správy spracované za zvolený posledný časový interval. Je to kvôli tomu, aby nespracovával viackrát tú istú RREQ správu prijatú od rôznych susedov. Uzol, ktorý vygeneroval a vyslal RREQ správu, čaká na odpoveď definovanú dobu a následne opakuje vyslanie novej RREQ. Toto môže opakovať až kým nedosiahne maximálny zvolený počet povolených opakovaní. Pri určovaní doby, po ktorú uzol čaká na doručenie odpovede na jednotlivé RREQ správy, by mala byť čakacia doba zvyšovaná exponenciálne. To znamená, že

čakacia doba každej vygenerovanej RREQ správy (okrem prvej) je nastavená na dvojnásobok čakacej doby predošlej RREQ správy. Počas toho, kým uzol čaká na odpoveď na jednotlivé RREQ správy, sú dátové pakety čakajúce na odoslanie hľadanému uzlu ukladané vo fronte čakajúcich paketov. Ak uplynula stanovená čakacia doba a uzol nedostal žiadnu odpoveď s cestou k hľadanému uzlu, buffrované pakety sú zahodené a vyššej vrstve je odoslaná správa o nedostupnosti cieľového uzla.

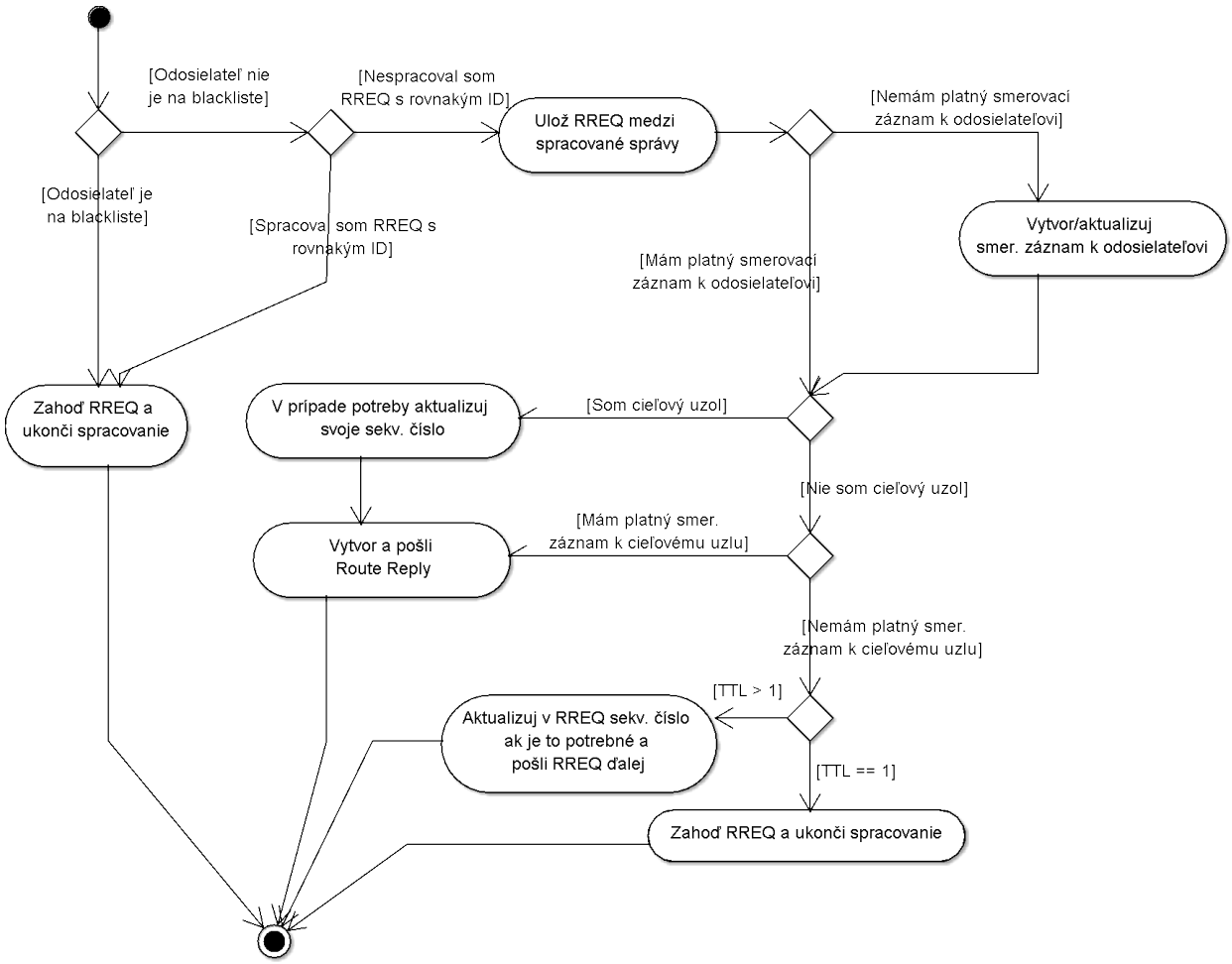


Obr. 3.2: Formát Route Request správy

RREQ správy sú vysielané broadcastom, tieto správy môžu prijať všetky susedné uzly s dostatočné silným signálom na príjem paketu.

3.1.2 Spracovanie prijatej Route Request

Uzol, ktorý prijal RREQ správu, si podľa prijatej správy vytvorí alebo aktualizuje v smerovacej tabuľke záznam k susednému uzlu, od ktorého RREQ prijal. Adresu uzlu si zistí z hlavičky IP paketu. Následne skontroluje podľa zoznamu spracovaných RREQ správ, či už rovnakú RREQ správu neprijal a nespracoval. V prípade, že už rovnakú RREQ správu spracoval, prijatú správu zahodí. V prípade, že takúto RREQ správu ešte nedostal, pokračuje v ďalšom spracovaní. Identifikáciu prijatej RREQ si uloží do zoznamu spracovaných správ, v smerovacej tabuľke vytvorí alebo aktualizuje smerovací záznam príslušný pôvodcovi RREQ správy a v prijatej správe zvýši hodnotu *Hopcount* o jedna. Vytvorený smerovací záznam k pôvodcovi správy uzol použije pre prípadné poslanie odpovede. Uzol si taktiež aktualizuje sekvenčné číslo pôvodcu RREQ na maximálnu hodnotu z aktuálne uloženého čísla a čísla uloženého v RREQ. Ako next hop zaznamená susedný uzol, od ktorého RREQ správu prijal. Ako vzdialenosť k tvorcovi RREQ je použitá hodnota Hopcount zaznamenaná v RREQ správe zvýšená o jedna. Doba platnosti tohto smerovacieho záznamu je nastavená na maximum z aktuálne uloženého času a času potrebného na propagáciu RREQ správy zvyšnou časťou siete.



Obr. 3.3: Spracovanie prijatej RREQ správy

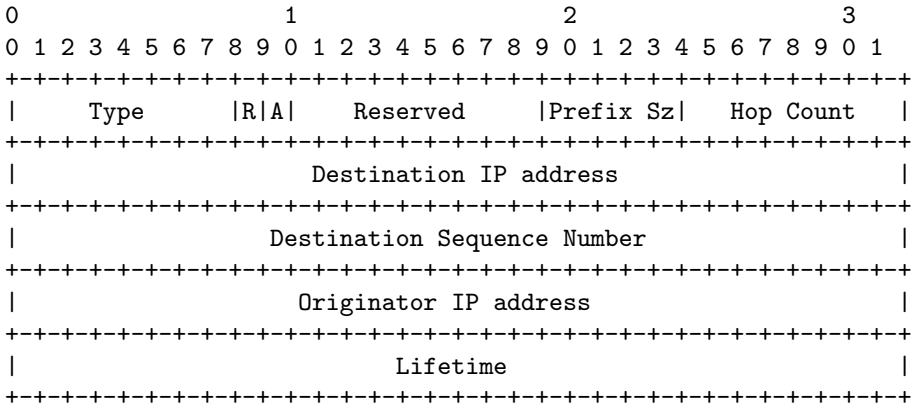
Po týchto krokoch uzol zistí, či môže poslať RREP. Kritéria určujúce či uzol môže alebo nemôže poslať odpoveď s cestou sú uvedené v nasledujúcom odseku. V prípade, že uzol nemôže poslať odpoveď a prijatá RREQ správa má v hlavičke IP paketu hodnotu *time to live* väčšiu ako 1, uzol broadcastom prepošle prijatú RREQ ďalej. Pred preposlaním uzol zníži hodnotu *time to live* o jedna a v RREQ správe ešte aktualizuje sekvenčné číslo hľadaného uzla na maximum z čísla uloženého v RREQ a čísla, ktoré ma pre daný cieľový uzol zaznamenané v smerovacej tabuľke, ak tam nejaké má uložené. Týmto sa zaručí, že uzol, ktorý vygeneroval RREQ správu, dostane najaktuálnejšiu informáciu o cieli. V prípade, že hodnota *time to live* v IP hlavičke prijatej RREQ už bola rovná 1, uzol RREQ ďalej neposiela a zahadzuje ju.

3.1.3 Generovanie Route Reply

Uzol, ktorý prijal RREQ správu, môže odpovedať zaslaním odpovede len v dvoch prípadoch:

1. v prípade, že je to uzol, ktorý je uvedený v RREQ správe ako cieľový uzol
2. v prípade platnosti nasledujúcich podmienok:
   - uzol má v smerovacej tabuľke aktívny záznam do hľadaného cieľového uzla
   - sekvenčné číslo cieľového uzla zaznamenané v smerovacej tabuľke je validné (známe) a nie je menšie ako číslo uvedené v RREQ správe
   - v prijatej RREQ správe nie je nastavený *Destination only* príznak, ktorý značí že na RREQ správu s týmto príznakom smie odpovedať len cieľový uzol a nie aj medziuzly, ktoré majú platnú smerovaciu informáciu o trase k cieľovému uzlu

Postup pri vytváraní RREP správy sa líši podľa toho, či uzol, ktorý generuje RREP, je cieľovým uzlom alebo tzv. medziuzlom. Medziuzlom nazývam uzol, ktorý sa nachádza na trase medzi zdrojovým uzlom a cieľovým uzlom a nie je to ani jeden z týchto krajných uzlov.
Ak odpoveď vytvára cieľový uzol, tak v prípade, že sekvenčné číslo cieľového uzla z RREQ správy je zhodné s jeho vlastným sekvenčným číslom, uzol zvýši svoje vlastné číslo. Ak je číslo v RREQ správe menšie ako jeho vlastné sekvenčné číslo, tak uzol svoje sekvenčné číslo nezvyšuje. Do vytváranej odpovede vloží svoje aktualizované sekvenčné číslo a hodnotu *Hopcount* v odpovedi nastaví na 0. Ak odpovedajúci uzol nie je cieľový uzol ale je to medziuzol, vtedy



Obr. 3.4: Formát Route Reply správy

vloží do odpovede sekvenčné číslo, ktoré má uložené v smerovacom zázname príslušného cieľového uzla. Do zoznamu uzlov, ktoré má označené ako uzly komunikujúce s cieľovým uzlom pridá susedný uzol, od ktorého prijal RREQ správu. K uzlom zaznamenaným ako tie, ktoré komunikujú s pôvodcom RREQ, pridá susedný uzol, ktorý je uvedený ako next hop na ceste k cieľovému uzlu. Hodnotu *Hopcount* nastaví na hodnotu, ktorú má uloženú v smerovacom zázname cieľového uzla.
Zvyšný postup je rovnaký či je uzol posielajúci odpoveď cieľový uzol alebo je to medziuzol. Do vytváranej RREP uzol vloží z RREQ správy adresu pôvodcu RREQ a sekvenčné číslo

pôvodcu uvedené v RREQ správe. Vytvorená RREP správe je následne poslaná unicastom po spätnej ceste smerom k pôvodcovi RREQ.
Označenie uzlov v odpovedi je rovnaké ako v RREQ správe. Ako cieľový uzol sa do RREP správy uvádza uzol, do ktorého bola hľadaná cesta - to je cieľový uzol z RREQ správy na základe ktorej je vytvorená odpoveď. Ako zdrojový uzol sa do RREP uvádza pôvodca RREQ správy.
V prípade, že bol v RREQ nastavený príznak *Gratitutious Route Reply* musí ešte uzol po odoslaní RREP správy odoslať ďalšiu RREP správu a to uzlu, ktorý bol v RREQ uvedený ako cieľový uzol. Deje sa to kvôli tomu, aby aj cieľový uzol získal informáciu o ceste k pôvodcovi RREQ správy. Bez tejto dodatočnej RREP by sa cieľový uzol nemusel dozvedieť o trase k pôvodcovi RREQ, a to v prípade, že na RREQ správu odpovedal medziuzol. Informácia o trase k pôvodcovi RREQ môže byť pre cieľový uzol dôležitá napríklad v prípade obojsmerného prenosu správ alebo v prípade vyžadovanej odpovede pre pôvodcu RREQ správy.

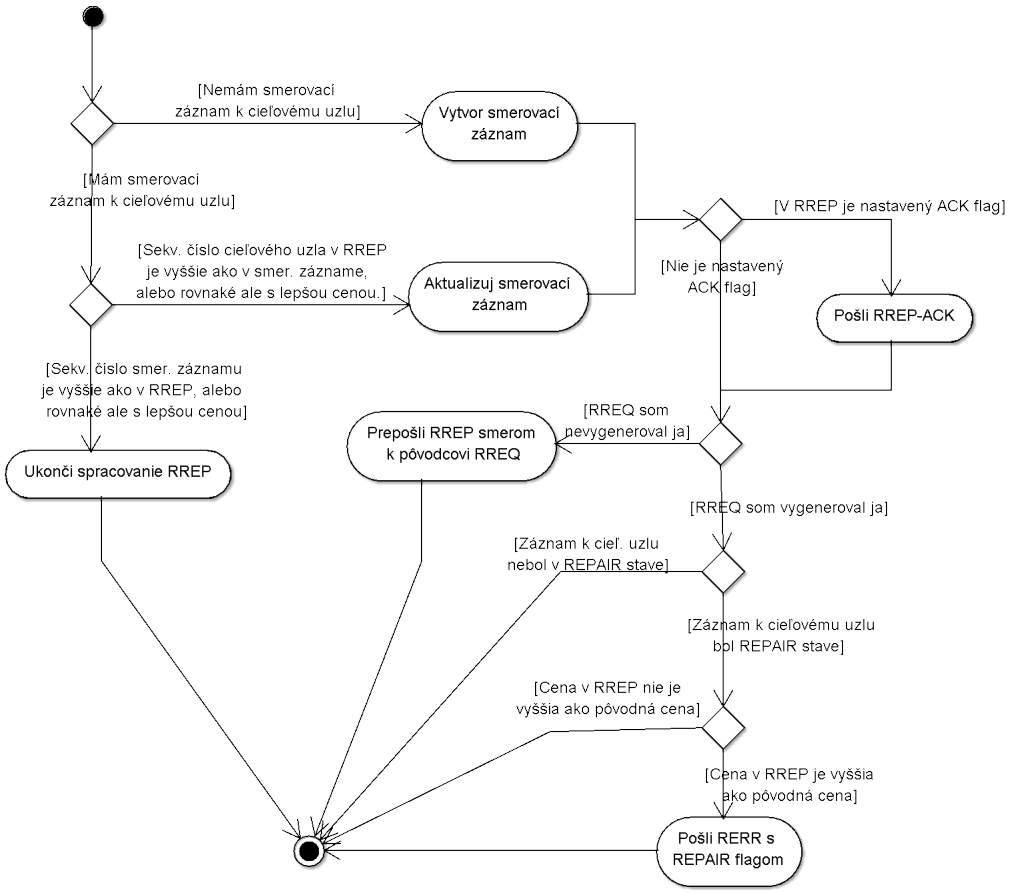
3.1.4 Spracovanie prijatej Route Reply správy

Uzol po prijatí RREP správy skontroluje, či v smerovacej tabuľke existuje smerovací záznam príslušný susednému uzlu, z ktorého bola prijatá RREP správa. Ak daný záznam neexistuje, uzol ho vytvorí. Následne uzol inkrementuje hodnotu *Hopcount* v RREP správe o jedna a v prípade, že v smerovacej tabuľke neexistuje záznam odpovedajúci uzlu, ktorý je v RREP uložený ako cieľový, uzol daný záznam vytvorí a uloží do neho odpovedajúce hodnoty z RREP (sú to sekvenčné číslo cieľového uzla, hopcount, next hop a doba platnosti smerovacieho záznamu).

Ak už daný záznam v smerovacej tabuľke existuje, uzol v ňom aktualizuje uvedené údaje ale len v prípade splnenia niektorej z nasledujúcich podmienok:

- sekvenčné číslo záznamu v smerovacej tabuľke je neznáme
- sekvenčné číslo je známe a je menšie ako sekvenčné číslo uvedené v RREP správe
- sekvenčné čísla sú rovnaké, ale smerovací záznam je označený ako neaktívny
- sekvenčné čísla sú rovnaké, ale hodnota *Hopcount* uložená v smerovacom zázname je väčšia ako hodnota *Hopcount* v RREP správe

Ak prebehla aktualizácia smerovacieho záznamu, záznam je označený ako platný a aktívny. Ako next hop je označený susedný uzol, od ktorého bola prijatá RREP správa. Ak uzol, ktorý prijal RREP správu nie je uzol, ktorému bola odpoveď určená, prepošle prijatú odpoveď ďalej k adresátovi. Adresu nasledujúceho uzla, kam má byť RREP správa preposlaná, by mal mať uzol uložený v smerovacej tabuľke z predošlého spracovania RREQ správy. Uzol následne pridá do zoznamu uzlov komunikujúcich s cieľovým uzlom susedný uzol, ktorému preposlal RREP. Do zoznamu uzlov komunikujúcich s pôvodcom RREQ uzol pridá susedný uzol, od ktorého prijal RREP. V prípade, že uzol preposiela RREP po spoji, ktorý považuje za nespoľahlivý, môže v RREP správe nastaviť príznak A. Tento príznak zabezpečí, že sused po prijatí RREP správy pošle naspäť potvrdenie o tom, že správu prijal.



Obr. 3.5: Spracovanie prijatej RREP správy

3.1.5 Zaistenie lokálnej konektivity

Každý uzol by mal mať aktuálne informácie o dostupnosti susedných uzlov s ktorými má vytvorené aktívne spojenie, to je od ktorých alebo ktorým preposiela dátové správy. Uzol zúčastňujúci sa na komunikácii musí mať možnosť overiť, že susedný uzol, ktorému preposiela správy je stále v dosahu vysielania a je schopný prijať požadovanú komunikáciu. K overeniu tejto dostupnosti je možné použiť viacero mechanizmov:

Prvým mechanizmom môže byť použitie odozvy z nižšej (linkovej) vrstvy. Ak linková vrstva poskytuje mechanizmus na overenie úspešnosti vyslania a príjmu správy susedným uzlom, je možné využiť túto možnosť. Príkladom sú napríklad potvrdzovacie správy u štandardu IEEE 802.11, kde uzol po prijatí unicastovej správy vysiela pre odosielateľa správy potvrdenie o jej úspešnom priatí. U broadcastových správ sa tieto potvrdenia nepoužívajú.

Ďalším možným mechanizmom je použitie Hello správ. Uzly vysielajú tieto správy v pravidelných intervaloch, počas ktorých nevyslali žiadnu broadcastovú správu. Každý uzol si ukladá čas, kedy vyslal poslednú broadcastovú správu. Ak počas uplynutia definovaného intervalu od posledného vyslania Hello správy uzol nevyslal žiadnu broadcastovú správu, vyšle

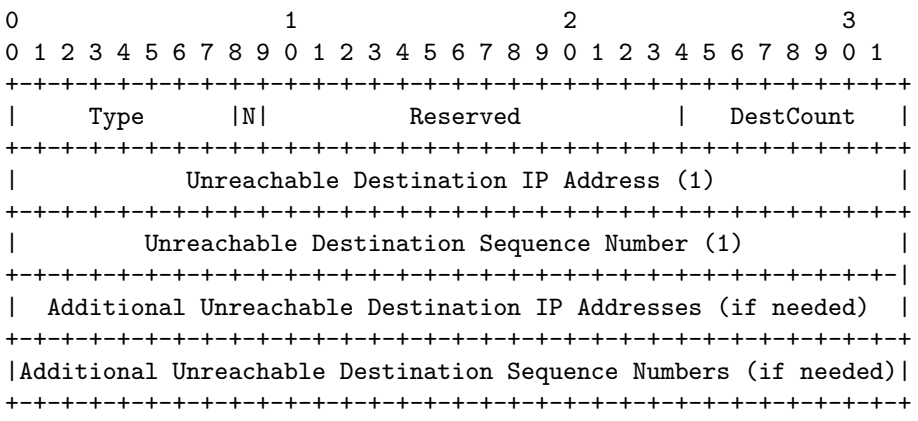
Hello paket. Ak počas definovaného intervalu uzol vyslal nejakú broadcastovú správu, uzol posunie vyslanie Hello správy na čas, ktorý bude odpovedať zvolenému intervalu pre vysielanie Hello správ od okamihu vyslania poslednej broadcast správy. Pred vyslaním Hello správy uzol uvedie do správy ako adresu cieľového uzla svoju vlastnú adresu, do správy vloží svoje aktuálne sekvenčné číslo a hodnotu *Lifetime* nastaví na interval, po ktorý môžu byť smerovacie záznamy ostatných uzlov vzťahujúce sa k tomuto uzlu považované za aktívne. V hlavičke IP paketu uzol nastaví hodnotu *Time To Live* na 1, čo zabezpečí, že Hello správa prijatá susednými uzlami nebude propagovaná ďalej. Ak uzol počas zvolenej doby neprijme od susedného uzla žiadnu Hello správu, považuje spojenie s týmto uzlom za prerušené. Uzol buď prevedie opravné operácie a pokúsi sa prerušené spojenie obnoviť alebo označí smerovací záznam ako neplatný a po uplynutí určitej doby záznam odstráni zo smerovacej tabuľky za predpokladu, že počas tejto doby nevznikla žiadna požiadavka použiť tento záznam.
V prípade ak použitá nižšia vrstva neposkytuje informačný mechanizmus a nie sú použité ani Hello správy, vysielajúci uzol môže podľa AODV návrhu použiť pasívne potvrdzovanie. Mechanizmus pasívneho potvrdzovania funguje na princípe sledovania prenosového média po určitý čas po vyslaní správy. Ak uzol očakáva, že adresát mal prijatú správy poslať ďalej a po určitú dobu nezaznamená od daného uzla žiadne vysielanie, považuje prenos správy za neúspešný. V takejto situácii môže použiť na overenie spojenia s daným uzlom vyslanie RREQ správy, prípadne iného kontrolného paketu, na ktorý má hľadaný uzol odpovedať. V prípade RREQ správy vysielajúci uzol uvedie ako hľadanú lokalitu uzol, s ktorým overuje spojenie. Ak sa hľadaný uzol stále nachádza v okolí vysielajúceho uzla, mal by vysielajúci uzol na zaslanú RREQ alebo iný kontrolný paket obdržať v krátkom čase od cieľového uzla odpoveď. Týmto je možné skontrolovať, či sú susedné uzly stále v dosahu vysielania.

3.1.6 Chyby v spojeniach

Uzly používajú na oznámenie o chybe v spojení správy typu Route Error (RERR). Uzol môže vygenerovať RERR správu ak nastala jedna z uvedených možností:

- ak uzol zistí prerušenie spojenia s uzlom, ktorý má v smerovacom zázname uložených nejakých predchodcov
- ak uzol prijme dátový paket určený adresátovi, pre ktorého nemá v smerovacej tabuľke záznam
- uzol prijme RERR správu od susedného uzla

Ak nastal prvý prípad, uzol do vygenerovanej RERR správy uloží adresy všetkých uzlov, do ktorých preposielal správy pomocou uzla, s ktorým sa prerušilo spojenie. To sú všetky uzly, ktoré majú v smerovacej tabuľke uzla ako next hop uvedený uzol, s ktorým sa prerušilo spojenie. Uzly, ktoré by mali obdržať RERR správu sú všetci predchodcovia uzlu s prerušením spojením. Ak nastal druhý prípad, nedostupný uzol je len jeden a to uzol, ktorý bol uvedený ako adresát správy. Adresu tohto uzla vloží uzol do RERR správy. V tomto prípade je príjemca RERR správy len jeden, a to susedný uzol od ktorého uzol prijal nedoručiteľnú správu.



Obr. 3.6: Formát Route Error správy

Pri prijatí RERR správy musí uzol skontrolovať všetky nedostupné adresy uvedené v RERR správe. Ak má v smerovacej tabuľke uložené záznamy príslušné adresám uvedeným v RERR a ako next hop má uložený susedný uzol, z ktorého prijal RERR, všetky takéto záznamy musí spracovať, aktualizovať a ich predchodcom poslať správu o prerušení spojenia. Pred poslaním RERR správy uzol aktualizuje záznamy v smerovacej tabuľke. Pre každý záznam z RERR správy uvedený ako nedostupný uzol v prvom a druhom prípade zvýši sekvenčné číslo záznamu. V treťom prípade uzol do záznamu v smerovacej tabuľke uloží sekvenčné číslo, ktoré bolo uložené v RERR správe. Následne uzol označí takýto smerovací záznam ako neplatný. Uzol týmto záznamom aktualizuje dobu platnosti, po ktorú sú neplatné záznamy udržované v tabuľke pred zmazaním na hodnotu aktuálneho času zvýšeného o interval *DELETE_PERIOD*.

3.1.7 Mechanizmus lokálnych opráv

V prípade chyby v spojení môžu uzly voliteľne použiť mechanizmus lokálnej opravy. Ak uzol detekuje chybu v spojení so susedným uzlom, ktorému preposiela správy, neposiela ihneď chybovú správu o výpadku spojenia ale snaží sa najprv samostatne opraviť prerušené spojenie. Uzol v takomto prípade zvýši sekvenčné číslo cieľového uzlu s prerušením spojením a vyšle RREQ správu, ktorá je propagovaná do určitého lokálneho okolia uzla. Ak uzol obdrží validnú odpoveď na RREQ správu, opraví si v smerovacej tabuľke záznam a týmto má obnovenú trasu k cieľovému uzlu. Pri aktualizácii smerovacieho záznamu uzol ešte skontroluje vzdialenosť k cieľovému uzlu, ktorú dostal v RREP so vzdialenosťou, v ktorej bol cieľový uzol pred výpadkom. Ak je vzdialenosť novej trasy väčšia, ako bola vzdialenosť predošlá, uzol vygeneruje RERR správu, v ktorej nastaví príznak `N` a pošle túto správu predchodcom uloženým v smerovacom zázname. Takto sa predošlé uzly dozvedia o zmene vzdialenosti a uzol, ktorý inicioval spojenie sa môže rozhodnúť, či pošle novú RREQ na nájdenie lepšej trasy alebo bude používať aktuálnu trasu. Uzol musí počas toho, ako čaká na odpoveď, na základe ktorej môže opraviť prerušené spojenie, ukladať doručené pakety, ktoré momentálne nemá kam preposlať. Ak po uplynutí definovanej doby uzol nedostane odpoveď ohľadom prerušeného spojenia, posiela RERR správu štandardným spôsobom popísaným v predchádzajúcej

kapitole. Pakety, ktoré zatiaľ uzol prijal a mal preposlať ďalej, uzol zahodí.
Uzol, ktorý prijme RERR správu s nastaveným príznakom `N`, neoznačí príslušný smerovací záznam, ktorého sa správa týka, ako neplatný. Uzol si podľa záznamov v správe aktualizuje sekvenčné čísla príslušných záznamov, zistí si predchodcov, ktorým by mal správu ďalej preposlať a RERR pošle ďalej. Takto je správa preposielaná až kým sa nedostane k uzlu, ktorý inicioval vyhľadanie trasy.

Mechanizmus lokálnych opráv môže zvýšiť pomer doručených a vyslaných paketov, zároveň ale opravené trasy môžu mať vyšší počet medziuzlov oproti optimálnym trasám. Na takýchto trasách rastie pravdepodobnosť chybného prenosu, môžu vznikať vyššie prenosové zdržania prípadne sa môže zvýšiť celkové vyťaženie siete. Výhodou zase je, že pri lokálnej oprave je RREQ správa vyslaná len v blízkom okolí uzla, kde sa trasa prerušila. Ak sa trasu podarí opraviť, zamedzí sa šíreniu RREQ až zo zdrojového uzla, čo môže zahrnúť podstatne väčšiu časť siete. Taktiež sa pri úspešnej oprave predíde strate paketov, ktoré už zdrojový uzol vyslal počas výpadku v trase.

3.1.8 Optimalizácia šírenia RREQ v sieti

Uzly môžu pri vyslaní RREQ správ do siete voliteľne používať techniku na obmedzenie vzdialenosti, do ktorej sú RREQ správy propagované. Táto možnosť má zamedziť zahlteniu celej siete častými RREQ správami. Mechanizmus predpokladá, že uzly komunikujú častejšie s uzlami, nachádzajúcimi sa v ich blízkosti ako s uzlami, ktoré sa nachádzajú sa vo väčšej vzdialenosti. V AODV protokole sa tento mechanizmus nazýva *Expanding Ring Search*. Pri prvom vyslaní RREQ správy uzol nastaví v IP hlavičke zvolenú hodnotu TTL. V závislosti na tejto hodnote uzol nastaví aj čas, ktorý čaká na príjem odpovede. Ak do uplynutia času odpoveď nedostane, vyšle novú RREQ so zvýšenou hodnotou TTL. Ak uzol nedostal odpoveď, postup opakuje až kým hodnota TTL nie je vyššia ako stanovený limit. Pri dosiahnutí stanoveného limitu uzol nastaví hodnotu TTL v nasledujúcej vysielanej RREQ správe na priemer celej siete a RREQ už nie je propagovaná len v ohraničenom okolí uzlu ale v rámci celej siete. V prípade, že uzol vyhľadáva trasu k uzlu, ktorého smerovací záznam má ešte uložený v smerovacej tabuľke, použije ako prvotné nastavenie TTL hodnotu zo smerovacieho záznamu zvýšenú o určitú vzdialenosť. Týmto uzol prehľadá v prvej fáze okolie, v ktorom by sa mohol uzol, s ktorým sa prerušilo spojenie, nachádzať s vyššou pravdepodobnosťou.

Kapitola 4

Modifikácie AODV s použitím alternatívnych metrík

Existuje množstvo typov alebo verzií smerovacích protokolov, ktoré vznikli úpravou alebo odvodením z protokolu AODV. Niektoré z takýchto verzií sú napríklad protokoly *AODV-JR* [15], *AODV-BR*[9] a *R-AODV* [26].
AODV-JR vzniklo zjednodušením AODV protokolu a vynechaním zložiek, ktoré podľa autorov neboli nutne potrebné pre fungovanie protokolu. Z pôvodného protokolu sú vynechané Hello správy, sekvenčné čísla, ohodnotenie trás počtom uzlov na trase a týmto zmenám je prispôsobené fungovanie protokolu. Ako najlepšie trasy sú zvolené trasy s minimálnym oneskorením medzi koncovými uzlami. Na vyslanú RREQ smie odpovedať len cieľový uzol, nie aj medziuzly, ako je to u AODV. Doba platnosti smerovacieho záznamu je aktualizovaná len pri príjme správy. Kvôli tejto zmene si musia koncové uzly na každej trase medzi sebou posielať kontrolné správy. Na základe tohto bolo možné odstrániť Hello správy a Route Error správy, ktoré už nie sú potrebné. V propagovanej RREQ správe nie je potrebné ukladať počet uzlov, ktorými správa prešla, lebo cieľový uzol odpovie len na prvú prijatú RREQ správu, to je správa s najmenším zdržaním. Výsledky získané pomocou simulácie ukázali výkon zrovnateľný s originálnou verziou protokolu.

Verzia AODV-BR je viaccestná verzia AODV protokolu (tzv. multipath), kde uzly využívajú informácie o súčasných viacerých cestách k cieľovému uzlu za účelom zvýšenia spoľahlivosti. V tejto verzii uzly promiskuitne zachytávajú všetky správy typu Route Reply prenášané v ich dosahu. Z týchto správ si uzly uložia smerovaciu informáciu k cieľovému uzlu, ku ktorému sa Route Reply správa vzťahuje. Po zostavení trasy prebieha komunikácia po trase, po ktorej bola posielaná Route Reply správa. V prípade, že sa vyskytne chyba v spojení medzi dvoma uzlami na trase, uzol z prerušeného spoja nachádzajúci sa bližšie k zdrojovému uzlu vysiela dátové správy broadcastom k svojim susedom. Susedné uzly môžu tieto správy preposielať ďalej k cieľovému ulzu, ak majú platnú smerovaciu informáciu. Uzol z prerušeného spoja zároveň vyšle Route Error správu smerom k zdrojovému uzlu, aby uzol inicioval nové vyhľadanie trasy. Uzly nachádzajúce sa v okolí zostavenej trasy si udržujú platné smerovacie informácie promiskuitným príjmom dátových paketov vysielaných po hlavnej trase. Výsledky simulačných testov ukázali zlepšenie v pomere doručených a vyslaných paketov oproti pôvodnému protokolu v redšie obsadených sieťach. V sieťach s vyšším počtom uzlov boli výsledky tejto

modifikácie mierne horšie.

V [26] je popísaná verzia protokolu Reverse AODV, ktorá vznikla na základe pozorovania nedoručených Route Reply správ. Pri strate Route Reply správy musí uzol, ktorý inicioval vyhľadanie trasy počkať kým uplynie zvolený čas a vyslať Route Request správu opäť. Týmto dochádza k zvyšovaniu zdržania, prípadne k zahadzovaniu buffrovaných dátových paketov. V práci je navrhnutý a odtestovaný spôsob zasielania Route Reply správ pomocou broadcastového vysielania namiesto unicastového. Všetky Route Reply správy sa pomocou broadcastu šíria k adresátovi viacerými trasami. Týmto sa zvýši šanca úspešného doručenia odpovede a zníži pravdepodobnosť potrebného znovu-vyslania Route Request správy. Výsledky prevedených testov ukázali zlepšenie oproti AODV protokolu v počte doručených správ ako aj v oneskorení, ktoré vzniklo počas prenosov správ.

Uvedené práce menia spôsob fungovania protokolu v relatívne väčšom rozsahu, ja som v tejto práci chcel ponechať rovnaký spôsob funkčnosti AODV protokolu a skúmať verzie s alternatívnymi metrikami, ktoré by brali pri ohodnocovaní do úvahy rušenie v sieti a stabilitu jednotlivých trás. Oproti pôvodnej verzii boli vykonané len zmeny bezprostredne potrebné k fungovaniu s novou metrikou.

Metrika je funkcia, ktorá priradzuje trase ohodnotenie definujúce aká dobrá je daná trasa. Na základe zvolenej metriky algoritmus vyberá cesty, ktoré bude používať k prenosu správ. Ohodnotenie alebo cena celej trasy obvykle vznikajú na základe kombinácie cien jednotlivých spojov, z ktorých sa trasa skladá. Väčšinou je toto celkové ohodnotenie určené ako suma ohodnotení jednotlivých zložiek trasy alebo ako minimálna (maximálna) hodnota niektorej zložky trasy, ale používajú sa aj iné spôsoby určenia ceny výslednej trasy z cien jednotlivých spojov trasy.

Najčastejšie používanou metrikou je ohodnotenie trasy podľa počtu uzlov, ktoré sa na danej trase nachádzajú. Táto metrika ale v bezdrôtových a v ad-hoc sieťach nie je úplne ideálna, pretože do ohodnotenia nezahŕňa prenosovú rýchlosť trasy, kvalitu trasy alebo veľkosť rušenia nachádzajúceho sa v okolí trasy.
Ak je výber trasy vykonaný len na základe počtu uzlov na trase, v niektorých prípadoch môže byť priepustnosť podstatne nižšia ako by bola pri výbere optimálnej trasy [19][18]. Tento efekt je častý v prípadoch, kedy v sieti existuje viac trás s rovnakým počtom uzlov, ale kvalitatívne sa trasy medzi sebou líšia. Porovnaním vlastností prenosov po trasách s krátkymi a dlhými spojmi sa zaoberá aj [23]. V bezdrôtových sieťach môžu mať spoje medzi uzlami na väčšiu vzdialenosť nižšiu kvalitu ako spoje na kratšiu vzdialenosť, prenos po takomto spoji vyžaduje viac energie a na takomto spoji existuje aj vyššia pravdepodobnosť chyby počas prenose. S viacerými uzlami zase rastie oneskorenie, ktoré vzniká spracovaním správ u každého uzla a stúpa pravdepodobnosť, že sa niektorý spoj preruší. Z [23] vyplýva, že nie je obecne možné jednoznačne určiť, či sú lepšie trasy s viacerými uzlami a menšou vzdialenosťou medzi sebou, alebo menším počtom uzlov nachádzajúcich sa vo väčších vzdialenostiach. Pre výber optimálnej trasy je potrebné zohľadniť viacero kritérií. Protokol AODV používa ako metriku práve počet uzlov na trase. Vzhľadom k vyššie uvedeným dôvodom som chcel navrhnúť a vybrať

iné metriky použiteľné s AODV protokolom a zistiť vplyv jednotlivých metrík na výkon siete.

Pre výber optimálnych trás používajú smerovacie protokoly grafové algoritmy. Najčastejšie používanými algoritmami sú Dijkstrov algoritmus a Bellman-Fordov algoritmus. Aby boli tieto algoritmy schopné nájsť na základe zvolenej metriky optimálnu trasu v rozumnom (polynomiálnom) čase, musí byť daná metrika *izotonická* [37]. Izotonicita je vlastnosť, ktorá zaručí, že vzájomné usporiadanie ohodnotení dvoch trás ostane rovnaké, ak sa pred obidve trasy pridá spoločná trasa alebo sa spoločná trasa pridá za obidve ohodnocované trasy.

Nech $W(x)$ je funkcia, pomocou ktorej metrika ohodnocuje trasy. Metrika je izotonická, ak pre ľubovoľné cesty a, b, c a c' platí ak $W(a) \leq W(b)$, potom $W(a \oplus c) \leq W(b \oplus c)$ a zároveň $W(c' \oplus a) \leq W(c' \oplus b)$, kde $\oplus$ je operátor zreťazenia príslušných ciest.

Izotonicita metriky je nutná a postačujúca vlastnosť, aby boli Dijsktrov a Bellman-Fordov algoritmus schopné určiť optimálnu trasu v polynomiálnom čase [37]. V opačnom prípade, ak metrika nie je izotonická, je potrebný exponenciálny čas na určenie optimálnej trasy pomocou uvedených algoritmov. Keďže vyhľadávanie najkratších ciest v AODV protokole prebieha na základe Bellman-Fordovho algoritmu, budú v tejto práci uvažované len izotonické metriky.

Prvé tri uvedené metriky sa budú zaoberať vplyvom rušenia a hustotou uzlov v sieti, nasledujúca metrika ETX je uvedená ako referenčná, aby bolo možné porovnať hodnoty s inou už existujúcou metrikou. Posledné tri metriky sa zaoberajú vplyvom stability spojov na výkon smerovacieho protokolu a prenosov v sieti.

Uzly v bezdrôtových sieťach komunikujú pomocou zdieľaného média, takže vysielanie jedného uzla spôsobuje rušenie ostatných zariadení, nachádzajúcich sa v dosahu tohto vysielania. Zariadenia komunikujúce na základe štandardu 802.11 a použitia prístupovej metódy CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) môžu vysielať v danom okamihu len v prípade, že v ich blízkosti nevysiela žiadne iné zariadenie. Uzol musí pred vysielaním detekovať médium po určitú dobu voľné a až potom môže odvysielať data. Následne po odvysielaní správy uzol čaká na potvrdenie o prijatí správy. Ak uzol detekuje pred plánovaným vysielaním vysielanie iného uzla, musí svoje vysielanie odložiť, počkať zvolenú dobu a po uplynutí tejto doby opakovať popísaný postup. S rastúcou hustotou vysielajúcich uzlov v sieti sa znižuje priepustnosť pre jednotlivé uzly. Rovnaký vplyv na priepustnosť siete má zvyšovanie počtu medziuzlov na trasách, po ktorých sú prenášané správy medzi koncovými uzlami. Napríklad pri komunikácii uzlov priamo medzi sebou (dĺžka trasy je rovná 1) a komunikácii s použitím jedného prostredníka (dĺžka trasy je rovná 2), sa zníži priepustnosť toku oproti prvému prípadu o 50%. Je to z dôvodu, že v jednom okamihu môže vysielať len jeden zo susedných uzlov trasy, buď zdrojový uzol alebo spomínaný prostredník.
Eliminácia rušenia je jedným z cieľov algoritmov zo skupiny *Topology Control*[35],[13]. Predpokladom použitia týchto algoritmov je, že uzly sú schopné redukovať vysielací výkon. Optimálnym znížením výkonu ovplyvní vysielanie uzlov menší počet susedných uzlov, čo má za následok menšie rušenie, viac uzlov môže vysielať súčasne a uzly takto taktiež šetria energiu.

4.1 MAODV

4.1.1 Popis

Z prác [22] a [24] vyplýva, že hlavným dôvodom nízkej priepustnosti v bezdrôtovej sieti je rušenie vznikajúce prenosom okolitých uzlov a viac-skokový (multihop) charakter prenosov v ad-hoc sieti. Ako prvá metrika, ktorá ma napadla a ktorú som chcel otestovať, je metrika minimalizujúca počet uzlov nachádzajúcich sa v okolí vybranej trasy. Predpokladom použitým pri návrhu tejto metriky je, že ak uzly budú vyberať trasy tvorené z uzlov, v ktorých okolí sa nachádza čo najmenší počet susedných uzlov, rušenie vzniknuté počas každého prenosu pokryje najmenšiu možnú časť siete a ostatné uzly budú mať možnosť vysielať v prípade potreby. Podľa tohto predpokladu by pri súčasnom vysielaní viacerých uzlov v sieti mal vznikať menší počet kolízií, uzly by sa mali dostať rýchlejšie k voľnému médiu a signál vyslaný jednotlivými uzlami by mal mať lepšiu kvalitu. Pri tejto metrike sa uzly snažia minimalizovať negatívne dôsledky vysielania po vybranej trase na zvyšnú časť siete, neberú do úvahy len svoj prospech. MAODV je názov pre verziu protokolu, kde je ako ohodnotenie trasy použitý súčet všetkých uzlov, nachádzajúcich sa v dosahu vysielania uzlov na trase medzi zdrojovým a cieľovým uzlom. Ohodnotenie je tým lepšie, čím menší je tento súčet.

Na podobnom princípe pracuje napríklad protokol *Average Link Interference-aware Routing* [40] navrhnutý na základe protokolu DSR, ktorý vyberá trasy s najnižšou priemernou interferenciou jednotlivých spojov trasy. Interferencia spoja je definovaná ako aritmetický priemer interferencií uzlov tvoriacich spoj. Interferencia uzla je určená ako počet susedných uzlov vynásobených váhou určenou podľa vzdialenosti, v ktorej sa konkrétny susedný uzol nachádza. Celkové ohodnotenie trasy je určené ako súčet interferencií všetkých spojov trasy vydelené počtom týchto spojov. S touto metrikou sa protokol snaží vyberať trasy vyhýbajúce sa oblastiam s vysokou hustotou okolitých uzlov.
Ďalším protokolom zahŕňajúcim ako kritérium pri výbere trás interferenciu je protokol *NAVC-driven AODV* [27]. Tento protokol je založený na AODV protokole a na informácii z MAC vrstvy, konkrétne na hodnote uloženej v *Network Allocation Vector* (NAV). V tomto vektore si MAC vrstva ukladá prehľad o obsadenosti a voľnom čase na prenosovom médiu. Do tohto vektoru si uzol zaznamenáva informácie na základe prijatých RTS a CTS správ, pomocou ktorých si uzly rezervujú médium, keď chcú vysielať. Na základe zaplnenia NAV je uzol klasifikovaný ako vyťažený alebo nie a podľa toho je zahrnutá jeho hodnota do ohodnotenia trasy. Na základe tejto metriky sú vyberané trasy skladajúce sa z uzlov s nízkym obsadením prenosového média.
Podobná metrika je navrhnutá aj v [34], ktorá k ohodnoteniu spojov používa informácie o čase, ktorý strávi MAC vrstva uzla v stave od začiatku pokusu o vysielanie až po úspešné vyslanie správy. Takýmito stavmi sú stav kedy uzol kontroluje obsadenosť média, stav kedy uzol detekoval kolíziu a čaká na uvoľnenie média alebo stav pred vyslaním dát, kedy musí uzol počkať náhodné zvolenú dobu kým môže dáta vyslať. Z súčtu týchto časov a celkového času od začiatku pokusu o vyslanie po jeho ukončenie je vypočítaný pomer, ktorý je použitý spolu s informáciou o šírke pásma daného spoja na určenie ohodnotenia daného spoja.

Vo verzii s názvom MAODV navrhnutej v tejto práci je formálne ohodnotenie trasy medzi uzlami A a B skladajúcej sa z uzlov $a_0..a_n$, kde $A = a_0$ a $B = a_n$ definované ako

$$\sum_{i=1..n-1} D_{a_i}$$

kde D_a definuje stupeň uzla a, čo je počet uzlov nachádzajúcich sa v dosahu jeho vysielania. Algoritmus sa snaží nájsť a vybrať trasu s minimálnym súčtom stupňov všetkých uzlov trasy. Do ohodnotenia trasy sa nepočítajú stupne počiatočného a koncového uzla trasy a_0 a a_n - tieto uzly budú vysielať bez ohľadu na vybranú trasu, takže počet uzlov v dosahu vysielania týchto koncových uzlov je rovnaký v prípade výberu ľubovoľnej trasy. Táto vlastnosť je platná len v prípade, že vysielací výkon koncových uzlov je fixne nastavený a nemení sa v závislosti na vybranej trase. Ak by sa vysielací výkon menil podľa vybranej trasy, je potrebné zarátať do ohodnotenia aj stupeň koncového uzla pri danom výkone. V prípade jednosmernej komunikácie sa jedná len o stupeň zdrojového uzla, keďže cieľový len prijíma a nič nevysiela. Informáciu o počte susedných uzlov si uzly aktualizujú zo správ prijatých od okolitých uzlov.

4.1.2 Implementácia a zmeny v protokole

Prvou potrebnou zmenou bolo pridanie dátovej položky pre evidenciu počtu susedov u každého uzla. V tejto položke si každý uzol udržuje počet aktívnych susedov, od ktorých prijal za posledný *ACTIVE_ROUTE_TIMEOUT* interval minimálne jednu kontrolnú správu. Do štruktúry reprezentujúcej smerovacie záznamy bolo potrebné pridať položku, v ktorej bude uložené ohodnotenie trasy príslušnej danému smerovaciemu záznamu.

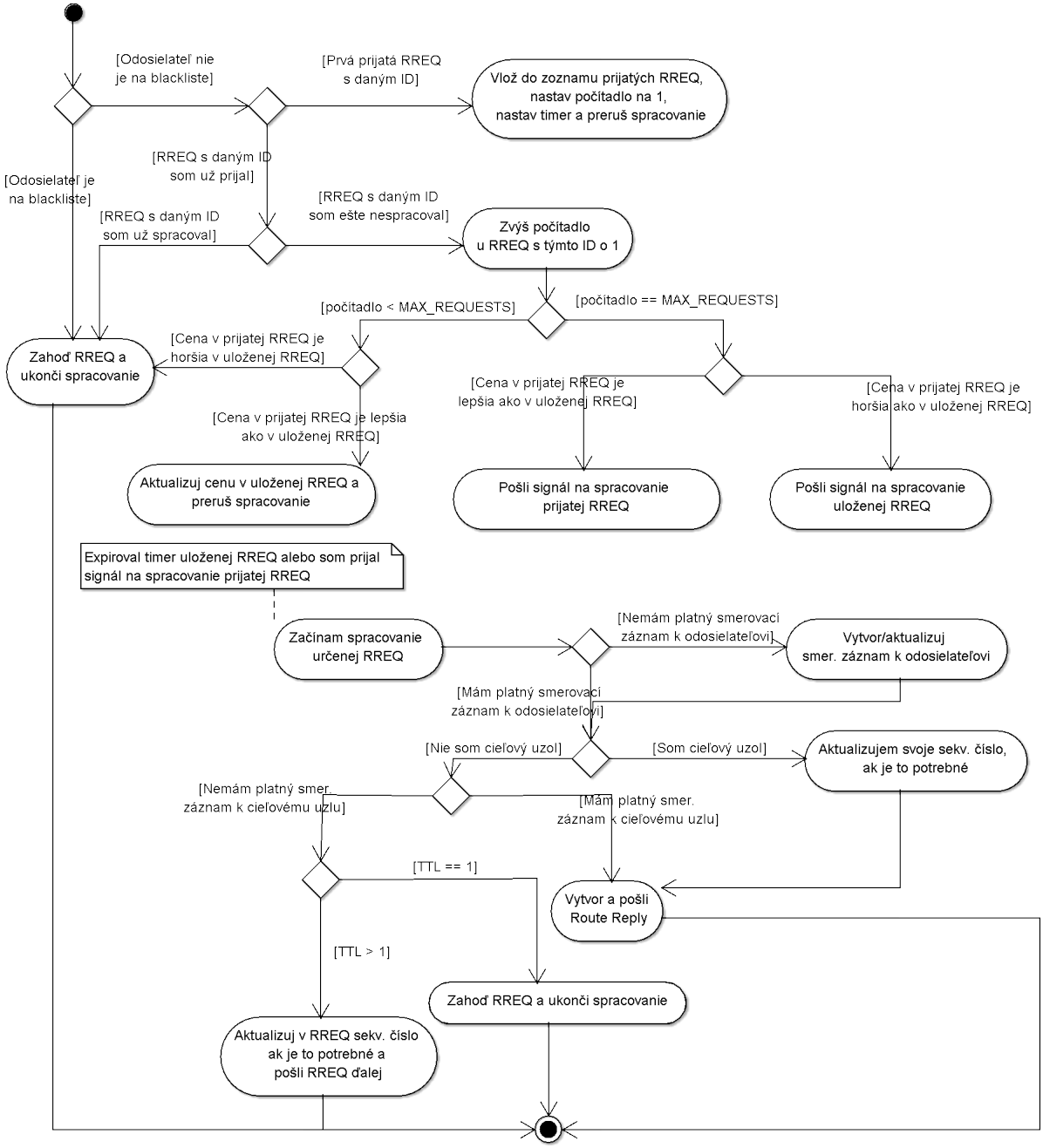
Oproti pôvodnému protokolu bolo potrebné zmeniť štruktúru RREQ a RREP správ. Do tried reprezentujúcich RREQ a RREP správy bola pridaná premenná, v ktorej je uložená cena trasy, ktorou bola zatiaľ správa preposlaná. V pôvodnej verzii protokolu je táto hodnota uložená v premennej s hodnotou hopcount, v tejto verzii protokol obsahujú správy obidve premenné - to je aj premennú s hopcount aj premennú s ohodnotením trasy. Hodnotu Hopcount je potrebné ponechať pre použitie pri lokálnych opravách alebo pri použití Expanding Ring Search, aby bolo možné určiť vzdialenosť, v akej sa uzol nachádzal.

Ďalšie zmeny bolo potrebné vykonať pri spracovaní RREQ správ. Prvou zmenou je, že uzol po prijatí RREQ správy najprv skontroluje, či už nespracoval správu s rovnakým ID. Ako ID správy je myslená dvojica $\langle adresa\ pôvodcu, route\ request\ ID \rangle$, ktorá jednoznačne identifikuje RREQ správu. Ak už rovnakú správu uzol spracoval, spracovanie tejto správy ukončí. Ak ešte rovnakú správu nespracoval, skontroluje či nemá prijatú RREQ správu preposlať ďalej a zároveň je hodnota TTL v správe rovná 1. Ak by nastal uvedený prípad, uzol správu zahodí a ukončí spracovanie správy rovnako ako v predošlom bode. V tomto prípade uzol zahodí správu ešte pred spracovaním preto, lebo na túto správu nemôže odpovedať a mal by ju preposlať ďalej, ale to nemôže kvôli hodnote TTL v IP hlavičke paketu, ktorá by bola po znížení nulová. Ak uzol správu nezahodil, môže nastať jeden z dvoch prípadov:

- prvým prípadom je, že uzol ešte neprijal RREQ správu s daným ID. V tomto prípade uzol uloží správu do zoznamu prijatých RREQ správ, kde je správa uložená definovanú dobu pred ďalším spracovaním, nastaví timer na zvolenú hodnotu a dočasne ukončí spracovanie.

- druhý prípad je, že uzol už prijal RREQ správu od rovnakým ID, ale zatiaľ ešte správu

nespracoval. V tomto prípade uzol nájde prijatú RREQ správu s daným ID v zozname prijatých správ čakajúcich na spracovanie, porovná hodnoty z uloženej a novoprijatej správy a ak je ohodnotenie novoprijatej správy lepšie, uzol aktualizuje hodnoty v uloženej správe hodnotami z novoprijatej správy.



Obr. 4.1: Spracovanie prijatej RREQ správy v modifikovaných verziách

Po uplynutí zvoleného intervalu uzol zo zoznamu vyberie uloženú správu a spracuje ju postupom zhodným s pôvodným algoritmom. Týmto postupom sa zabezpečí, že spracovávaná

RREQ správa je najlepšia správa z RREQ správ s rovnakým ID prijatých za definovaný interval. Negatívnym efektom je mierne zvýšenie oneskorenia, ktoré vznikne čakaním správy pred samotným spracovaním. Aby správa obsahovala počet uzlov v okolí trasy po ktorej bola správa posielaná, každý uzol, ktorý danú správu práve preposiela, pripočíta k hodnote uloženej v správe počet svojich susedov.

4.2 TRMAODV

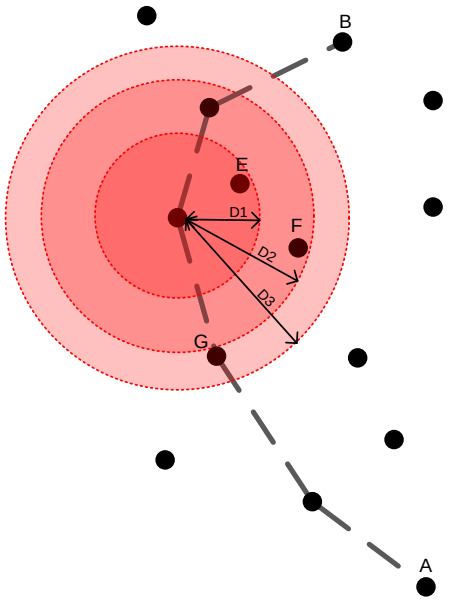
4.2.1 Popis

Druhou metrikou, ktorá ma napadla a berie do úvahy rušenie v sieti je metrika použitá vo verzii s názvom TRMAODV. U predošlej verzie MAODV uzol zahrnul do ohodnotenia trasy počet všetkých svojich susedných uzlov. Takéto ohodnotenie môže byť skreslené počtom susedných uzlov, ktoré nič nevysielajú alebo sa nepokúšajú o vysielanie. Uzly, ktoré nič nevysielajú, nie sú zdrojom rušenia a ani nie sú obmedzované vysielaním ostatných okolitých uzlov. V tejto verzii sa trasy ohodnocujú na základe objemu vysielania, ktoré je prenášané cez jednotlivé uzly trasy a taktiež vysielané v okolí týchto uzlov. Kritéria, ktoré by sa mali zlepšiť oproti pôvodnej verzii sú rovnaké ako u predošlej modifikácie MAODV, teda zvýšenie pomeru úspešne doručených správ oproti vyslaným správam a zníženie oneskorenia vzniknutého u doručených správ.
V tejto verzii si každý uzol počíta objem dátových správ, ktoré vyslal za posledný stanovený interval. Do tohto počtu sa počítajú správy vygenerované samotným uzlom ako aj správy preposlané, ktoré uzol prijal od okolitých uzlov a preposlal ďalej. Pri vyslaní Hello správy uzol vloží objem vyslaných správ do pripravovanej správy, čím sa dozvedia všetky susedné uzly, akým objemom vysielania prispieva tento uzol k vyťaženiu prenosového média a tým aj celkovému rušeniu. Pri príjme Hello správy si prijímajúci uzol zo správy vyberie uloženú hodnotu o vysielanom objeme a uloží ju k smerovaciemu záznamu uzla, od ktorého správu prijal. Keďže uzol pri vyslaní RREQ správy odloží vyslanie Hello správy, vkladá uzol informáciu o vyslanom objeme dát aj do RREQ správ, aby mali okolité uzly aktuálne a pravidelné informácie o počte vysielaných správ.

K samotnému ohodnoteniu trás potom každý uzol používa hodnotu

$$all\_sent\_packets\_size = \alpha * packets\_size\_sent\_by\_me + packets\_size\_sent\_by\_neighbors$$

kde prvá zložka je objem správ, ktoré vyslal daný uzol a druhá zložka je súčet objemu správ vyslaných okolitými susedmi. α je koeficient, ktorým je vynásobený objem paketov zaslaných samotným uzlom. Keďže okolité uzly sa môžu od daného uzla nachádzať v rôznych vzdialenostiach, ich vysielanie prispieva k rušeniu rôznymi úrovňami. Uzly v bližšom okolí konkrétneho uzla spôsobujú svojím vysielaním tomuto uzlu väčšie rušenie ako vzdialenejšie uzly. Preto má aj každý susedný uzol pridelený koeficient - váhu, ktorým sa násobí objem jeho vysielania. Tento koeficient sa určí na základe vzdialenosti, v ktorej sa susedný uzol nachádza. Uzly sú podľa vzdialenosti rozdelené do 3 kategórií (na obrázku 4.2 sú označené ako D1, D2 a D3), každá kategória ma pridelenú váhu, ktorá sa znižuje s rastúcou vzdialenosťou.



Obr. 4.2: Výber trasy a priradenie váh susedným uzlom u verzie TRMAODV

Vzdialenosť je určená zo sily signálu, s akou je prijímané vysielanie od susedného uzla. Pri výpočte vzdialenosti je použitý *two-ray ground* model šírenia rádiového signálu, podľa ktorého je sila signálu prijatého vo vzdialenosti d od bodu vyslania určená vzťahom

$$P(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L}$$

kde

- P_t - je sila, s akou bol signál vyslaný
- G_t - je zisk (zosilnenie) antény uzla vysielajúceho signál
- G_r - je zisk antény uzla prijímajúceho signál
- L - je koeficient straty vo voľnom priestore prostredia
- h_t je výška, v ktorej je umiestnená anténa vysielajúceho uzla
- h_r je výška, v ktorej je umiestnená anténa prijímajúceho uzla

Pri výpočte uzol predpokladá, že všetky uzly majú rovnako nastavené parametre vysielača rádiového signálu.

4.2.2 Implementácia a zmeny v protokole

Evidovanie odoslaných objemov je realizované pomocou premennej, ktorá slúži ako počítadlo odoslaných objemov za časovú jednotku a pomocou poľa, do ktorého sú ukladané v pravidelných intervaloch hodnoty tejto premennej. Pri každom vyslaní dátovej správy je do spomínanej premennej pripočítaná veľkosť odoslaného paketu. V zvolených intervaloch (defaultne 1 sekunda) sa hodnota tejto premennej uloží do poľa na nasledujúcu pozíciu a premenná sa vynuluje. Indexy ukladaných pozícií sú určované cyklicky cez rozsah celého poľa. Celková hodnota odoslaných objemov za sledovaný interval je daná súčtom všetkých položiek popisovaného poľa.
Do dátových štruktúr reprezentujúcich Hello správy a RREQ správy je hodnota o objeme vyslaných správ vkladaná na koniec správy vo forme rozšírenia, ktoré je definované v [7] v kapitole 9. Do správ je vložená len samotná hodnota objemu vyslaných správ. IP adresu uzla, ku ktorému sa táto hodnota vzťahuje, získa prijímajúci uzol z hlavičky samotnej správy, preto túto adresu nie je potrebné do rozšírenia vkladať. V súvislosti s evidovaním odoslaných objemov bolo potrebné pridať k dátovým položkám uzla jeden timer, ktorý slúži pre periodické ukladanie hodnôt evidenčnej premennej do odkladacieho poľa ako aj na aktualizáciu celkového objemu odoslaných správ, ktorý sa vkladá do RREQ správ a Hello správ.
Do štruktúry smerovacieho záznamu boli pridané dve premenné slúžiace pre evidenciu odoslaných objemov. Do premennej *sent_packets* si uzol ukladá hodnoty o objeme vyslaných správ, ktoré získal od daného susedného uzla z prijatých RREQ a Hello správ. V druhej premennej s názvom $koeff$ si uzol uchováva váhu, s ktorou násobí získaný objem správ. Túto váhu uzol nastavuje v závislosti na vzdialenosti od daného uzla. Hodnoty obidvoch týchto premenných sú nastavované a majú zmysel len v prípade, ak je daný uzol uzlom susedným - nachádza sa vo vzdialenosti 1. V ostatných prípadoch smerovacích záznamov nie sú hodnoty týchto premenných využívané.

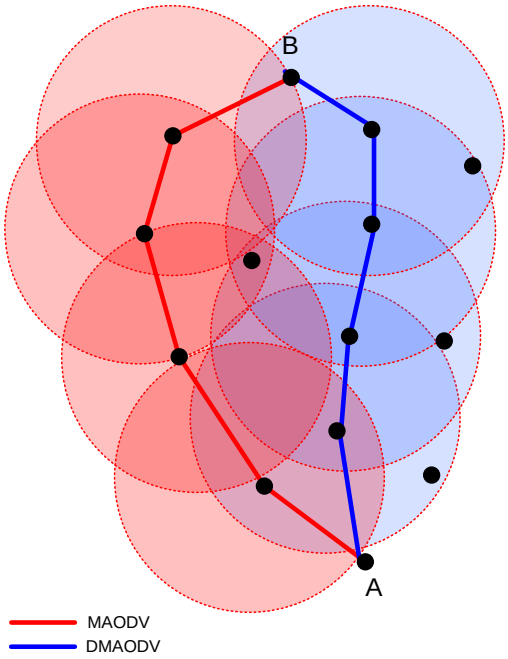
4.3 DMAODV

4.3.1 Popis

Verzia protokolu DMAODV vznikla ako modifikácia verzie MAODV. Metrika v tejto verzii je založená rovnako ako metrika u MAODV na stupni uzlov tvoriacich jednotlivé trasy. Algoritmus s touto metrikou sa ale nesnaží vyberať trasy skladajúce sa z uzlov s minimálnym počtom susedov, ale naopak vyberá trasy skladajúce sa z uzlov, ktoré majú čo najvyšší počet susedných uzlov.
Cieľom tejto metriky je použiť trasy, u ktorých je možné pri prípadnom výpadku čo najrýchlejšia oprava. Predpokladom je, že v prípade výskytu chyby v spojení medzi dvoma uzlami na vybranej trase je lokálna oprava prerušeného spojenia úspešná s vyššou pravdepodobnosťou a v kratšom čase, ak mali dané uzly vyšší počet susedných uzlov. Z výsledkov tejto a predošlej metriky je možné porovnať, aký vplyv má stupeň uzlov tvoriacich trasu na výkon celej siete. Formálne je ohodnotenie trasy medzi uzlami A a B skladajúcej sa z uzlov $a_0 ... a_n$, kde $A = a_0$ a $B = a_n$ definované ako

$$\sum_{i=1..n-1} \frac{1}{D_{a_i}}$$

kde D_a definuje stupeň uzla a. Algoritmus vyberá trasu s minimálnym ohodnotením zo všetkých dostupných ohodnotení.



Obr. 4.3: Výber trasy medzi bodmi A a B u verzií MAODV a DMAODV.

4.3.2 Implementácia a zmeny v protokole

Pre fungovanie algoritmu s popísanou metrikou je nutné, aby mali uzly informáciu o počte svojich susedov, tak ako to bolo u verzie smerovacieho protokolu MAODV. Za týmto účelom boli v pôvodnom smerovacom protokole vykonané rovnaké zmeny ako u verzie MAODV. Rovnaké je aj rozšírenie dátových štruktúr o položky potrebné pre prácu s počtom susedných uzlov. Aby nebolo nutné meniť funkcie pre spracovanie RREQ a RREP správ a aby fungoval algoritmus na vyhľadávanie najkratších ciest korektne, pri ohodnocovaní spojov sa používa prevrátená hodnota stupňa uzla, teda $1/D(a)$. Algoritmus takto vyberie trasu s minimálnym ohodnotením, čo odpovedá trase skladajúcej sa z uzlov s najvyšším stupňom. Okolnosti, za ktorých uzol aktualizuje počet susedných uzlov, sú rovnaké ako okolnosti u verzie MAODV.

4.4 EMAODV

4.4.1 Popis

EMAODV je názov pre verziu AODV protokolu, ktorá používa metriku Expected Transmission Count (ETX). Táto metrika bola navrhnutá p. De Couto [19]. Cieľom metriky je vyberať

trasy s najvyššou pravdepodobnosťou bezchybného prenosu medzi koncovými uzlami. Metrika sa snaží brať do úvahy pri ohodnocovaní nielen počet uzlov na danej trase, ale aj kvalitu a chybovosť spojov medzi uzlami. Hodnota spoja podľa tejto metriky znamená očakávaný počet prenosov na linkovej vrstve potrebných na úspešné doručenie jednej správy susednému uzlu. V tejto hodnote je započítaný prenos správy od vysielajúceho uzla k prijímajúcemu ako aj vyslanie potvrdenia o prijatí správy a úspešné prijatie tohto potvrdenia protiľahlým uzlom. Hodnota celej trasy je určená sumou hodnôt jednotlivých spojov, z ktorých sa trasa skladá. Na výpočet ocenenia spojenia medzi uzlami A a B sú použité hodnoty D_f a D_r, ktoré znamenajú pravdepodobnosť úspešného prenosu správy od uzla A k uzlu B a pravdepodobnosť úspešného prenosu správy v opačnom smere. Pri tejto metrike uzol periodicky vysiela broadcastom v definovaných intervaloch kontrolné správy. Každá broadcastová správa by mala byť vyslaná linkovou vrstvou len raz, u týchto správ sa nekontroluje, či boli úspešne prijaté okolitými uzlami alebo nie. Uzol si pre každý susedný uzol počíta hodnotu, koľko kontrolných správ prijal od tohto susedného uzla za určený časový interval. Tieto údaje sú použité na výpočet hodnôt D_f a D_r pre konkrétneho suseda. Z vypočítaných hodnôt je určené výsledné ohodnotenie spoja s daným susedným uzlom podľa vzťahu

$$ETX = \frac{1}{D_f \times D_r}$$

Pomer úspešne prijatých správ v čase t od susedného uzla vypočíta uzol podľa vzťahu

$$D_r(t) = \frac{count(t-w, t)}{w/\tau}$$

kde $count(t-w, t)$ znamená počet kontrolných správ prijatých od susedného uzla za interval w a w/τ znamená počet kontrolných správ, ktoré mal uzol za interval w od susedného uzla prijať. Táto metrika berie do úvahy aj asymetrické spoje, po ktorých sú správy úspešne doručované len v jednom smere. V prípade výskytu asymetrického spoja sa v jednom smere počet prijatých správ od susedného uzla rovná nule, čím sa zvýši celková hodnota spoja.

4.4.2 Implementácia a zmeny v protokole

Existujúcu implementáciu metriky ETX do AODV protokolu je možné nájsť v [8]. Návrh a realizácia niektorých častí ETX do protokolu sa mi nezdali v uvedenej verzii vyriešené ideálnym spôsobom, preto som implementoval vlastnú verziu.

Uzly musia pre výpočet spoľahlivosti jednotlivých spojov a výpočet ETX hodnôt periodicky vysielať kontrolné správy. Ako uvedené kontrolné správy sú použité Hello správy protokolu AODV, ktoré zároveň slúžia aj na overenie dostupnosti susedných uzlov. Uzol vkladá do Hello správy pre každý susedný uzol, ku ktorému má v smerovacej tabuľke platný záznam, IP adresu tohto uzla a hodnotu D_r. Tieto hodnoty sú vložené do Hello správy vo forme rozšírenia popísaného v kapitole 9 [7]. Rozdielom oproti pôvodnému protokolu je vysielanie Hello správ v pravidelných intervaloch. V pôvodnom protokole sa pri vyslaní broadcastovej správy posunie vyslanie Hello správy o patričný interval. V modifikovanej verzii protokolu sa toto posunutie nedeje z dôvodu, aby boli uzly schopné určiť počet kontrolných správ, ktoré

mali za definovaný interval prijať. Druhým rozdielom oproti pôvodnému protokolu je, že pri použití metriky ETX vysielajú Hello správy všetky uzly, nie len uzly nachádzajúce sa na aktívnej trase.
Pri tejto implementácii je do každého smerovacieho záznamu pridané pole o veľkosti počtu sekúnd sledovaného intervalu. Defaultne je tento interval nastavený na 10 sekúnd. Pri príjme Hello správy si uzol do poľa smerovacieho záznamu príslušnému uzlu, od ktorého prijal Hello správu, zaznamená čas príchodu Hello správy a príznak, že hodnota na danom indexe je platná. Z údajov uložených v tomto poli je uzol schopný pre daného suseda zistiť, koľko Hello správ od suseda prijal za posledný sledovaný interval. Pri vyslaní Hello správy uzol spočíta pre každého suseda koľko správ od neho za sledovaný interval prijal, zistí koľko správ mal prijať podľa času, ako dlho je smerovací záznam aktívny a vloží do pripravovanej správy pomer týchto hodnôt spolu s adresou príslušného susedného uzla. Ako dlho je smerovací záznam aktívny uzol zistí z časovej značky, ktorá je pridaná do smerovacieho záznamu. Do tejto časovej značky je uložený čas, kedy bol daný smerovací záznam nastavený ako platný.

Pri príjme Hello správy uzol z prijatej správy získa hodnotu kvality spojenia v smere od tohto uzla k susednému uzlu. Kvalitu spoja v opačnom smere, to je v smere od susedného uzla k aktuálnemu uzlu, zistí uzol spočítaním prijatých správ za sledovaný interval. Z týchto hodnôt uzol následne určí celkovú kvalitu spoja a aktualizuje ju v smerovacej tabuľke. Každý smerovací záznam naviac obsahuje timer, ktorý zabezpečí aktualizáciu ohodnotenia v smerovacej tabuľke v prípade neprijatia Hello správy počas definovaného intervalu. Čas tohto timeru je pri každom prijatí Hello správy nastavený na očakávaný čas príchodu nasledujúcej Hello správy.
Ďalšou zmenou oproti pôvodnému protokolu je formát RREQ a RREP. Pri prijatí RREQ alebo RREP správy uzol zvýši hodnotu ETX uloženú v správe o hodnotu spoja k susednému uzlu, od ktorého správu prijal. Následné spracovanie RREP správy je rovnaké ako u pôvodnej verzie protokolu. Pri spracovaní RREQ správ je použitý rovnaký princíp buffrovania prijatých správ ako u predchádzajúcich popísaných verzií a následné spracovanie týchto správ je rovnaké ako u pôvodnej verzie.

4.5 TMAODV

4.5.1 Popis

U mobilných sietí je jedným z dôležitých kritérií stabilita spojov, po ktorých sú realizované prenosy. Spoje môžu byť nestabilné z viacerých príčin, buď z dôvodu nízkej sily prijímaného signálu alebo nízkeho odstupu prijímaného signálu od hladiny rušenia v okolí, kedy je uzol schopný korektne prijať len niektoré vyslané správy, alebo z dôvodu mobility uzlov, keď sa jednotlivé uzly dostanú mimo dosah vzájomného vysielania. Nestabilné spoje spôsobujú vyššie oneskorenie pri prenose a zvýšené režijné náklady potrebné na opravu alebo vyhľadanie nových trás. Predikciou doby existencie spojov medzi uzlami sa zaoberajú práce [10] a [30]. V týchto prácach sú využívané informácie o pozícii uzlov alebo vzdialenostiach uzlov medzi sebou. Informácie o aktuálnej pozícii uzlov je možné získať napríklad pomocou GPS, v prípade použitia informácii o vzdialenostiach medzi uzlami je možné tieto vzdialenosti určiť výpočtom z hladiny prijímaného signálu. V [16] je analyzovaný výkon siete pri použití trás

skladajúcich sa z najmenšieho počtu uzlov a trás s najvyššou stabilitou. Stabilita je meraná časom do prvého prerušenia ľubovoľného spoja. Štúdia taktiež obsahuje návrh protokolu *Long Lifetime Routing* (LLR), pomocou ktorého je možné nájsť cestu s najvyššou životnosťou medzi dvoma uzlami. Doba existencie jednotlivých spojov je určovaná na základe rýchlosti a smeru, akou sa uzly pohybujú. Tento algoritmus je však určený len na vyhodnocovacie účely, ako vstup potrebuje údaje o celej sieti. Algoritmus nie je možné použiť na výber trás v samotných uzloch. Jedným z prvých smerovacích protokolov pre ad-hoc siete, ktorý uvažuje vo výbere trás stabilitu vybraných spojov je *Associativity Based Routing* protokol [38]. V tomto protokole sú trasy vyberané primárne podľa stability liniek, z ktorých sa trasa skladá. V prípade rovnosti tohto kritéria u viacerých trás je použité ako ďalšie kritérium počet uzlov na trase a oneskorenie vzniknuté pri šírení Route Request správy. U tohto protokolu si každý uzol ukladá informácie o stabilite spojenia so susednými uzlami, tieto informácie následne ukladá do vysielaných Route Request správ. Pre monitorovanie spojov so susednými uzlami vysiela každý uzol periodicky kontrolné správy.
Ďalšou verziou ktorá berie do úvahy stabilitu je obmena AOMDV protokolu s názvom *Stability Based Partially Disjoint AOMDV* [11]. Cieľom protokolu je vyberať trasy s najvyššou stabilitou. Stabilita spoja medzi dvoma uzlami je meraná počtom Route Request správ prijatých navzájom. Najvyššie ohodnotenie má trasa, ktorá ma najvyšší priemer stabilít všetkých spojov trasy. Pri viacnásobnej trase medzi zdrojovým a cieľovým uzlom protokol vyberá za účelom zvýšenia spoľahlivosti trasy, ktoré obsahujú najmenej spoločných uzlov. Primárnym kritériom je ale stabilita.

Pod názvom TMAODV som navrhol modifikáciu AODV protokolu používajúceho metriku, ktorá pri ohodnocovaní ciest používa časové kritérium dĺžky existencie jednotlivých spojov, z ktorých sa trasa skladá. Straty paketov pri použití AODV protokolu môžu byť zapríčinené z viacerých dôvodov. Prvou príčinou strát je nenájdenie trasy k cieľovému uzlu u uzla kde vznikla požiadavka na prenos. Počet týchto strát by nemal byť ovplyvnený výberom metriky. Ak existuje trasa medzi dvoma uzlami, mal by ju nájsť protokol s pôvodnou aj s alternatívnou metrikou. Druhou príčinou strát paketov je prerušenie existujúcej trasy, po ktorej sú práve vysielané správy. Počet týchto strát je možné výberom metriky ovplyvniť. Predpokladom metriky použitej vo verzii protokolu TMAODV je, že čím dlhšie existuje spojenie medzi dvoma uzlami, za tým stabilnejšie je možné dané spojenie považovať. Použitím stabilných spojení sa predíde častým výpadkom v prenosovej trase zapríčineným prenosom pomocou uzlov, ktoré majú vysoký stupeň mobility a na vybranej trase sa nachádzajú len krátku dobu.

Formálne je ohodnotenie ciest určené takto: Nech c je trasa z uzla A do uzla B, t.j. $c = a_0...a_n$, kde $a_0 = A$, $a_n = B$ a $a_{i,i>0,i<n}$ sú uzly danej trasy vedúcej z A do B. Nech $T(a, b)$ znamená čas vzniku spoja z uzla a do uzla b ako aj spoja z uzla b do uzla a. Tu je predpoklad, že spoj je symetrický a spoj vznikol v obidvoch smeroch približne v rovnakom čase (malý časový rozdiel je možné zanedbať). Potom hodnota cesty c je určená ako

$$H(c) = \max_{i=0...n-1} T(a_i, a_{i+1})$$

Vo funkcii H má z dvoch časov menšiu hodnotu ten, ktorý je na časovej osi umiestnený viac vľavo. Algoritmus pre výber ciest vyberie ako optimálnu cestu tú, ktorá ma najnižšie ohod-

notenie, to znamená, že najmladší spoj na danej ceste existuje dlhšiu dobu, ako najmladší spoj všetkých ostatných ciest. Výsledná váha trasy zloženej z viacerých spojov je určená najnižšou váhou spoja zo všetkých spojov trasy.

4.5.2 Implementácia a zmeny v protokole

Pri použití tejto metriky sú smerovacie záznamy rozšírené o časový údaj, ktorého hodnota predstavuje jeden z dvoch významov:

- v prípade, že smerovací záznam je záznam príslušný susednému uzlu, hodnota znamená čas, kedy bol smerovací záznam naposledy označený ako aktívny
- v prípade uzla s hodnotou $hopcount > 1$ hodnota znamená čas vzniku posledného spoja na danej ceste, to je spoja, ktorý existuje najkratší čas zo všetkých spojov trasy

Pri vzniku smerovacieho záznamu je časová značka nastavená na aktuálny čas v prípade ak uzol, ku ktorému sa viaže smerovací záznam, je susedný uzol. V prípade, že je to uzol vo vzdialenosti väčšej ako 1, časová značka je nastavená podľa hodnoty, ktorá bola uložená v prijatej RREP správe na základe ktorej záznam vzniká.

Typy správ RREQ a RREP boli rozšírené o číselnú hodnotu, ktorá znamená dobu existencie najkratšie existujúceho spoja na ceste, po ktorej správa bola zatiaľ správa posielaná. Uvedená doba znamená počet sekúnd, ako dlho spoj existuje. V správe je uložená doba v sekundách a nie konkrétny čas, ako dlho spoj existuje, kvôli tomu, aby jednotlivé uzly nemuseli mať zosynchronizované hodiny. Po prijatí RREQ alebo RREP správy uzol najprv zistí, ako dlho existuje spoj so susedným uzlom, od ktorého správu práve prijal. Hodnotu, ktorá je uložená v správe odpočíta od aktuálneho času a takto získa čas vzniku najmladšieho spoja. Tieto dve hodnoty uzol môže porovnať a neskorší čas z týchto dvoch uloží do prijatej správy. Následné spracovanie RREP správy prebieha tým istým postupom ako je to u pôvodnej verzie protokolu. Pri spracovaní RREQ správy sú prijaté správy buffrované určitú dobu, aby bolo možné preposlať ďalej alebo odpovedať na najlepšiu z prijatých správ. Spracovanie po vybratí najlepšej RREQ správy je rovnaké ako u ostatných verzií.

Pri použití tejto metriky vysielajú Hello správy všetky uzly v sieti, nie len tie, ktoré sa nachádzajú na aktívnych trasách. Vysielané Hello správy sú použité pre oznámenie o prítomnosti uzla potencionálnym susedným uzlom. Týmto spôsobom je možné u uzlov sledovať dĺžku existencie jednotlivých spojení s ich susednými uzlami.

4.6 SMAODV

4.6.1 Popis

Iným prístupom okrem predošlých uvedených k výberu dostatočne stabilných ciest je použitie informácií z prijímaného signálu. Parametrami, ktoré je možné použiť sú napríklad sila signálu, s akou uzol prijíma od susedných uzlov správy, odstup od šumu alebo chybovosť

prenášaných správ. Nevýhodou je premenlivý charakter týchto ukazateľov kvôli meniacim sa podmienkam okolitého prostredia, rušenia vznikajúceho viaccestným šírením signálu, pohybom prijímajúceho uzla alebo prekážkami v trase šírenia signálu. Aby mohol smerovací protokol, ktorý je v OSI modeli na 3 vrstve (sieťovej) využívať informácie o signále prijímaných správ, musí komunikovať aj s vrstvou fyzického rozhrania, nie len so susednou linkovou vrstvou. Typ návrhu protokolov, kde je využitý tento spôsob komunikácie, sa nazýva tzv. 'cross-layer design' [12], [36].
V [21] je popísaný upravený protokol *Destination Source Routing*, ktorý používa pre výber trasy miesto kritéria dĺžky trasy informácie o najnižšej hodnote odstupu signálu od rušenia zo všetkých spojov trasy a informáciu o najnižšej sile prijímaného signálu. Pri propagovaní Route Request správ uzly ukladajú tieto hodnoty do šírenej správy. Cieľový uzol na základe týchto hodnôt vyberie najlepšiu správu na ktorú odpovie. Prevedenými testami bolo ukázané zvýšenie priepustnosti siete a zníženie chybovosti vyberaných trás.
V [20] je navrhnutá obmena protokolu DSR, ktorý pri šírení Route Request používa prah pre hodnotu SNR. Uzol prepošle a spracuje len tie Route Request správy, u ktorých odstup signálu od šumu, pomocou ktorého boli prijaté, je vyšší ako nastavený prah. Hodnota prahu je nastavená empiricky podľa predošlých pozorovaní.
Ďalším protokolom používajúcim silu prijímaného signálu je protokol *Signal Strength Aware routing*(SSR)[29]. V tomto protokole uzol pri prijme Route Request správy na základe sily signálu, s akou bola táto správa prijatá, určí interval, za ktorý prijatú správu vyšle ďalej do siete a taktiež rýchlosť, s akou bude daná správa vyslaná. Správam prijatým po spojoch so silnejším signálom sú nastavené kratšie intervaly, čím je zabezpečené, že tieto správy sa dostanú rýchlejšie k cieľovému uzlu. Pre spojenie je takto vybraná cesta so stabilnejšími spojmi. Protokol používa mechanizmus *auto rate fallback* (ARF), pri ktorom uzol adaptívne určuje ako rýchlosťou bude vysielať správy.
Prah u hodnoty SNR je použitý aj v [28]. V práci sú analyzované tzv. "šedé zóny" pri použití AODV protokolu, čo sú okrajové oblasti dosahu vysielania uzlov. Komunikácia medzi uzlom a susednými uzlami nachádzajúcimi sa v jeho šedej zóne má vysokú pravdepodobnosť chýb počas prenosu. Jedným z riešení navrhnutých autormi výskumu bolo použitie prahovej hodnoty SNR u kontrolných Hello paketov. Ak je paket prijatý signálom, ktorý ma menší odstup od hladiny rušenia, ako je nastavená hodnota prahu, paket nie je ďalej spracovaný. Týmto sa zníži chybovosť prenosov, na druhej strane sa obmedzí konektivita v prípade neexistencie inej trasy k uzlu v šedej oblasti.

SMAODV je názov pre verziu protokolu, ktorej cieľom je minimalizovať počet nedoručených paketov. S metrikou minimalizujúcou počet uzlov na vyberanej trase môže algoritmus vyberať spojenia, ktoré tvoria uzly nachádzajúce sa od seba v relatívne veľkej vzdialenosti. Dôsledkom tejto vzdialenosti môže byť sila alebo kvalita prijatého signálu podstatne nižšia ako u uzlov nachádzajúcich sa blízko seba. Použitím týchto spojov môžu vznikať chyby pri prenose alebo môže klesať použiteľná rýchlosť prenosov. Signál prijatý z väčšej vzdialenosti má menšiu kvalitu, pri výskyte rušenia v okolí prijímajúceho uzla nemusí byť možné správne interpretovať správy z prijatého signálu. Taktiež pri uzloch nachádzajúcich sa vo väčšej vzdialenosti stačí pohyb menšieho rozsahu, aby sa uzly dostali mimo dosah vzájomného vysielania, čím dôjde k výpadkom spojenia na trase. Tieto výpadky v lepšom prípade zvyšujú oneskorenie pri prenose správ a vyššie množstvo režijných správ. V horšom prípade, ak sa prerušenú

trasu nepodarí opraviť a zdrojový uzol o tom nemá informáciu, spôsobia výpadky nedoručenie vyslaných správ. Verzia SMAODV ohodnocuje spojenia medzi jednotlivými uzlami podľa pomeru sily signálu a rušenia v okolí, s akým každý z týchto uzlov prijíma správy od protiľahlého uzla. Algoritmus s navrhnutou metrikou sa týmto okolnostiam snaží predchádzať preferovaním trás skladajúcich sa zo spojov, ktorých uzly medzi sebou prijímajú vysielanie s čo najvyššou silou. Ohodnotenie trasy je určené minimálnou hodnotou sily prijímaného signálu zo všetkých dvojíc susedných uzlov, z ktorých sa trasa skladá.
Ohodnotenie trasy c z uzla A do uzla B, t.j. $c = a_0...a_n$, kde $a_0 = A$, $a_n = B$ a $a_{i,0<i<n}$ sú uzly nachádzajúce sa na danej trase je určené ako

$$H(c) = \max_{i=0..n-1} \frac{1}{S(a_i, a_{i+1})}$$

kde, funkcia $S(a, b)$ vracia silu signálu s akou prijíma uzol b správy od uzla a. Algoritmus sa snaží vybrať trasu s minimálnym ohodnotením.
Pri súčasnom vysielaní viacerých uzlov uzol prijme správu len zo signálu, ktorý je zo všetkých prijatých signálov najsilnejší a len za podmienky, že rozdiel medzi silou tohto signálu a silou ostatných signálov je dostatočne veľký. Podľa použitej fyzickej a linkovej vrstvy a informácii, ktoré sú schopné tieto vrstvy poskytnúť, môže byť použitý ako indikátor kvality a sily signálu buď samotná sila prijatého signálu alebo odstup sily prijatého signálu od hladiny šumu v okolí uzla. Ak berieme do úvahy len silu signálu bez ohľadu na okolitý šum, je možné, že v okolí vybraných spojov bude vysoký šum, čím v konečnom dôsledku bude kvalita signálu horšia ako na inom možnom spoji s nižšou silou signálu ale minimálnou hodnotou šumu. Z tohto dôvodu je lepším ukazovateľom odstup signálu od šumu, kde sú zohľadnené obidve zložky ovplyvňujúce príjem správ. Fyzická a linková vrstva uzlov ale musia v tomto prípade umožniť sieťovej vrstve získať tieto hodnoty.

4.6.2 Implementácia a zmeny v protokole

Pri použití tejto metriky bolo potrebné rozšíriť dátové typy smerovacích záznamov o premennú, v ktorej bude uložená minimálna sila signálu na trase k cieľovému uzlu, ku ktorému sa viaže daný smerovací záznam. Táto hodnota znamená minimálnu silu signálu zo všetkých dvojíc uzlov na trase, s akou medzi sebou daná dvojica susedných uzlov komunikuje. Každý uzol si pri príjme dátovej správy alebo kontrolnej správy AODV protokolu od susedného uzlu zaznamená silu signálu, s akou danú správu prijal a premietne ju do ohodnotenia smerovacieho záznamu príslušného susedného uzla. Pri ukladaní hodnôt sily signálu je použitá metóda exponenciálne váženého kĺzavého priemeru, kde aktualizovaná hodnota je počítaná podľa vzťahu

$$cost = rxPr * \alpha + (1 - \alpha) * cost$$

kde *cost* je premenná s aktuálnym ohodnotením spoja, α je vhodne zvolený parameter z intervalu [0,1] a $rxPr$ je sila signálu aktuálne prijatej správy. Použitím uvedeného vzťahu sú v ohodnotení zahrnuté aj hodnoty sily signálu z predošlých správ. Čím je parameter α bližšie k hodnote 1, tým vyššia váha je v ohodnotení prikladaná hodnotám z nových správ a tým rýchlejšie strácajú v ohodnotení vplyv hodnoty z predošlých správ.

Oproti pôvodnému algoritmu je ďalej zmenená propagácia RREQ správ, pri ktorých je použité

buffrovanie správ pred spracovaním počas definovanej doby rovnako ako je to uvedené u predchádzajúcich modifikovaných verzií. Pri prijatí RREQ a RREP správy uzol skontroluje ohodnotenie uložené v prijatej správe a silu signálu s ktorou danú správu prijal. Hodnota uložená v správe znamená minimálnu hodnotu sily signálu, s akou bola daná správa prijatá na doterajšej ceste. Uzol z týchto dvoch hodnôt vyberie minimum, s ktorým pracuje v ďalšom spracovaní správy. Uzol taktiež aktualizuje hodnotu uloženú v správe na vybrané minimum. Následný postup spracovania je zhodný s postupom uvedeným v predošlých verziách.

4.7 SDMAODV

4.7.1 Popis

Poslednou metrikou ktorú som navrhol a implementoval v rámci tejto práce je metrika nazvaná SDMAODV, ktorá využíva kombináciu kritérií sily signálu, oneskorenia vzniknutého na trase a dĺžky trasy (myslené počtom uzlov na trase). Predchádzajúca metrika, ktorá ohodnocovala trasu minimálnou hodnotou signálu zo všetkých dvojíc uzlov na trase, produkovala v testovacích scenároch dobré výsledky, ale nebrala do úvahy prenosovú rýchlosť jednotlivých spojov. V testovacom prostredí vysielajú všetky uzly rovnakou rýchlosťou, v reálnom prostredí môžu niektoré uzly vysielať menšou rýchlosťou, prípadne môžu byť zahltené veľkým množstvom prenášaných správ, takže na trase obsahujúcej tento uzol môže vznikať vyššie oneskorenie. Z tohto dôvodu je ako dodatočné kritérium k sile signálu použitá aj hodnota oneskorenia na jednotlivých spojoch trasy. Tretím doplňujúcim kritériom tejto metriky je počet uzlov, z ktorých sa trasa skladá. Čím je počet uzlov vyšší, tým vyššia je pravdepodobnosť prerušenia spojenia trasy na niektorom zo spojov a s tým rastie aj pravdepodobnosť, že vyslaná správa nebude úspešne doručená.
Ohodnotenie trasy c z uzla A do uzla B, t.j. $c = a_0...a_n$, kde $a_0 = A$, $a_n = B$ a $a_{i,0<i<n}$ sú uzly nachádzajúce sa na danej trase je určené ako

$$H(c) = \alpha \min_{i=0..n-1} \frac{1}{S(a_i, a_{i+1})} + \sum_{i=0..n-1} \beta D(a_i, a_{i+1}) + \gamma n$$

kde, funkcia $S(a, b)$ vracia silu signálu s akou prijíma uzol b správy od uzla a a $D(a, b)$ vracia hodnotu oneskorenia, ktoré vznikne pri prenose správ z uzla a do uzla b. Pomocou koeficientov α, β a γ je možné meniť váhu jednotlivých kritérií v celkovom ohodnotení trasy. Najlepšia trasa je trasa s minimálnym ohodnotením.

4.7.2 Implementácia a zmeny v protokole

Pri implementácii bolo potrebné vykonať zmeny podobné ako u predošlej modifikácie, s tým že naviac bolo potrebné rozšíriť triedy predstavujúce RREQ a RREP správy o dátovú položku, v ktorej je ukladaná hodnota oneskorenia. Taktiež do záznamov smerovacej tabuľky bolo potrebné pridať položky používané pri meraní oneskorenia medzi susednými uzlami.
Pre počítanie hodnôt oneskorenia boli použité Hello správy, aby nebolo nutné vysielať k meraniu oneskorenia osobitné správy. Ideou navrhnutého spôsobu merania oneskorenia bolo, že uzol pri vyslaní Hello správy vloží do správy čas, kedy správu vyslal. Uzol, ktorý danú

správu prijme, si uloží k záznamu suseda časovú značku vloženú v správe ako aj čas, kedy Hello správu prijal. Následne keď nastane čas, aby tento uzol vyslal Hello správu, uzol vloží do Hello správy adresy všetkých uzlov, od ktorých má odložené časové značky. K týmto adresám vloží uložené časové značky navýšené o dobu, ktorá uplynula medzi prijatím Hello správy od príslušného susedného uzla až po aktuálny okamih, kedy bude uzol vysielať Hello správu. Po prijatí tejto Hello správy bude môcť susedný uzol zistiť hodnotu oneskorenia, ktoré vzniklo v prenose smerom k susednému uzlu a naspäť tým, že odčíta aktuálny čas od časovej značky, ktorú práve prijal. Týmto postupom nie je nutné, aby mali uzly v sieti synchronizované hodiny, lebo pre výpočet používajú čas, ktorý uložili do správy podľa svojich vlastných hodín. Problémom je, že doba u novovzniknutých susedných uzlov medzi zistením prvej hodnoty oneskorenia môže byť v najhoršom prípade až $2 \times HELLO\_INTERVAL$. Po túto dobu, kedy uzol nemá informáciu o reálnom oneskorení, používa uzol priemernú hodnotu oneskorenia vypočítanú z hodnôt uložených v záznamoch všetkých susedných uzlov. Aby nedochádzalo k pauzám v meraní oneskorenia zapríčineným odložením vyslanie Hello správ pri vyslaní inej broadcastovej správy, uzly vkladajú uložené časové značky aj do Route Request správ vo forme rozšírenia tejto správy. Takto dostávajú uzly aktuálne informácie častejšie, ako keby boli časové značky vysielané len v Hello správach. Ostatné zmeny v protokole boli rovnaké ako u predošlých uvedených verzií.

Kapitola 5

Nástroje pre simuláciu siete

5.1 Výber simulačného nástroja

Pre zistenie a porovnanie výkonnosti metrík, ktoré som sa rozhodol implementovať do AODV protokolu, bolo potrebné zvoliť vhodný nástroj, použitím ktorého by bolo možné porovnanie previesť. Na pokusné otestovanie a získanie výsledkov bolo potrebné vytvoriť sieť s desiatkami uzlov rozmiestnených na relatívne rozsiahlej ploche. Jednotlivé testovacie scenáre je potrebné vykonať viackrát s rôzne zvolenými parametrami. V reálnom prostredí by bolo toto prevedenie veľmi náročné, preto sa na experimentálne vyhodnocovanie používajú simulácie.
Pre simuláciu je potrebný model prostredia, ktoré má byť simulované. Simulačný model by mal odpovedať realite, aby výsledky, ktoré pomocou simulácie získame, odpovedali výsledkom dosahovaným v skutočnom prostredí. Čím viac má model odpovedať realite, tým detailnejšie musí byť model navrhnutý a prepracovaný. S vyšším stupňom detailu rastie náročnosť simulácie a počet operácií, ktoré je potrebné pri simulácii vykonať, preto je potrebné nájsť a zvoliť vhodný kompromis medzi detailom spracovania simulačného modelu a náročnosťou simulácie tohto modelu.
Ďalším krokom je výber nástroja, pomocou ktorého je možné prevádzať simuláciu zvoleného modelu. Nástrojov, ktoré sú schopné simulovať chod siete s bezdrôtovými mobilnými uzlami, je relatívne vysoký počet. Na začiatku bolo potrebné vybrať si z množstva dostupných nástrojov ten správny.
Pri výbere nástroja boli zohľadňované viaceré kritéria, hlavnými z nich boli:

- dostupnosť simulačného nástroja (voľne dostupný, voľne dostupný len pre akademické účely, platený)
- možnosti simulácie, ktoré daný model poskytuje
- hardwarové nároky a softwarové nároky daného nástroja
- aktuálnosť daného nástroja
- rozšírenosť simulačného nástroja a početnosť používateľskej komunity
- ohlasy ostatných užívateľov a stupeň podpory (rozsiahlosť a kvalita dokumentácie k nástroju, rozširujúce balíky, doplnky)

5.2 Dostupne simulačné nástroje

V nasledujúcej časti je uvedený prehľad najčastejšie používaných nástrojov pre simuláciu sieťových prostredí.

5.2.1 Network Simulator 2 (NS2)

NS2 je simulátor diskrétnych udalostí určený pre simuláciu rôznych typov sietí. Použitie je možné na systémoch s unixovým operačným systémom. Tento simulačný nástroj má za sebou pomerne rozsiahly vývoj a má pomerne stabilnú pozíciu v oblasti simulácií sietí. Vývoj prvej verzie, ktorá bola označená NS (Network Simulator), začal v roku 1989, od roku 1995 bol vývoj tejto verzie podporovaný známou agentúrou Defense Advanced Research Projects Agency (DARPA). V roku 1996 bola vydaná druhá verzia simulátoru s označením NS2. NS2 obsahuje podporu pre bezdrôtové a senzorové siete, simuláciu real-time komunikačných systémov, množstvo prenosových, smerovacích a aplikačných protokolov. Architektúra prostredia odpovedá návrhu OSI architektúry. Simulované uzly sa skladajú zo sieťových vrstiev, ktoré si medzi sebou predávajú správy. Inštalácia NS2 obsahuje komponenty umožňujúce simuláciu LAN, WLAN a satelitných sietí. V nástroji je implementované množstvo smerovacích protokolov, či už statických alebo dynamických protokolov s podporou unicastu alebo multicastu. Z modulov aplikačnej vrstvy sú v nástroji štandardne k dispozícii TCP, UDP a RTP protokoly. Okrem modulov, ktoré sú zahrnuté v štandardnom inštalačnom balíku NS2 existuje množstvo ďalších rozšírení implementovaných rôznymi používateľmi NS2, ktoré si je možné do prostredia NS2 doinštalovať. NS2 je vyvíjaný ako open-source software, čo umožňuje početnej komunite používateľov prispievať k rozvoju a zlepšovaniu nástroja. Pri implementácií sú použité 2 jazyky, prvým je objektové rozšírenie skriptovacieho jazyka Tcl nazývané OTcl. Druhým použitým jazykom je C++. V C++ je napísané jadro simulačného nástroja, ktoré sa stará o beh samotnej simulácie a moduly, ktoré predstavujú sieťové protokoly alebo rozhrania. Konfiguračná časť je napísané v jazyku Object-oriented Tool Command Language (OTcl). V jazyku OTcl používateľ nadefinuje konfiguráciu samotnej simulácie, v C++ používateľ implementuje chovanie modulov a podmodulov, z ktorých sa skladajú simulačné uzly. NS2 neposkytuje používateľom pre ovládanie behu simulácie žiadne grafické rozhranie, simulácia sa púšťa pomocou príkazového riadku. Počas simulácie je možné zaznamenávať priebeh simulácie na rôznych vrstvách do súboru a pomocou externých programov následne spracovať. Pre grafické zobrazenie zaznamenaného priebehu simulácie a získaných výsledkov sú dostupné nástroje napríklad Network Animator (NAM), Trace graph alebo INSpect. Súbor s priebehom simulácie má textovú formu, na spracovanie je možné použiť aj utility ako sú awk alebo bash skripty. NS2 je jeden z najviac používaných nástrojov v oblasti simulácii sieťových protokolov. Výhodou NS2 je open-source GPL licencia, početná komunita používateľov využívajúcich tento nástroj a z toho vyplývajúci dobrý stupeň podpory. NS2 obsahuje taktiež rozsiahlu dokumentáciu ako aj množstvo tutoriálov oboznamujúcich používateľov s použitím NS2.

5.2.2 OMNeT++

OMNeT++ je open-source nástroj, v ktorom je možné simulovať širokú škálu procesov. Je voľne dostupný pre akademické a neziskové použitie. Nástroj funguje na platformách typu OS Unix, Linux, MAC OS a Windows. Pre komerčné použitie existuje verzia OMNEST označovaná ako OMNeT Enterprise Edition. Vývoj OMNeT++ začal v roku 1992 vývoj na University of Budapest študentom Adrasom Vargom. Nástroj je napísaný v jazyku C++, podľa ktorého je aj pomenovaný. Jadro OMNeTu obsahuje implementáciu starajúcu sa o simuláciu diskrétnych udalostí, sieťové vrstvy a protokoly sú doplnené vo forme externých frameworkov, ktoré využívajú simulačné funkcie jadra OMNeTu. Vzhľadom k tomu je tento nástroj možné použiť nielen v spojení so sieťovými technológiami, ale na ľubovoľný učel, ktorý vyžaduje použitie simulátora diskrétnych udalostí. OMNeT používa komponentovú architektúru. Systém, ktorý sa má simulovať, sa skladá z hierarchicky usporiadaných modulov alebo komponent. Jednotlivé moduly sú implementované v C++ jazyku. Z týchto modulov sú pomocou jazyka NED (Network Description Language) vytvárané zložitejšie moduly a komponenty, ktorých vstupy a výstupy je možné prepájať s vstupmi a výstupmi ostatných komponent. Pomocou týchto prepojení si moduly medzi sebou predávajú správy. OMNeT obsahuje integrované vývojové prostredie založené na platforme Eclipse. Pre koncových používateľov OMNeT ponúka grafické prostredie, v ktorom je možné ovládať simuláciu a detailne sledovať jej priebeh. Simulácie je možné taktiež spúšťať jednotlivo alebo dávkovo z príkazového riadku, bez použitia grafického prostredia. Výsledky chodu simulácie je možné ukladať do súborov, ktoré je možné použiť na ďalšie spracovanie. Nástroje na štatistické spracovanie týchto súborov sú obsiahnuté priamo v inštalácii OMNeTu. Taktiež je možné zozbierané dáta exportovať do formátov použiteľných externými nástrojmi ako sú Matlab alebo iné.

5.2.3 GloMoSim (Global Mobile System Simulator)

GloMoSim mal byť podľa oficiálnej stránky [4] nástroj určený pre simuláciu rozsiahlych káblových ako aj bezdrôtových sietí, aj keď v dostupných verziách podporuje len simuláciu bezdrôtových sietí. Doplnenie možnosti simulácie klasických káblových sietí bolo plánované v nových verziách, ale nebolo uskutočnené. Použitie nástroja je bezplatné pre vzdelávacie účely, výskum alebo neziskovú činnosť. Vývoj je zastavený približne od roku 2000, kedy bola vydaná posledná verzia. Na základe GloMoSimu je vyvíjaná komerčná verzia s názvom QualNet. GloMoSim je knižnica modelov určená pre jazyk Parsec (Parallel Simulation Environment for Complex Systems), čo je simulačný jazyk určený pre simulácie diskrétnych udalostí postavený na jazyku C. V GloMoSime je použitá vrstvová architektúra podobná architektúre OSI. Jednotlivé simulačné uzly sa skladajú z vrstiev, ktoré majú definované jednotné API, pomocou ktorého komunikujú s ostatnými vrstvami daného uzla. Konfigurácia simulačného scenára sa zadáva vo forme textového súboru. Simulácia sa spúšťa z príkazového riadku, na zobrazenie priebehu simulácie je v GloMoSime nástroj GloMoSim Visualizatin Tool. Z protokolov sieťovej vrstvy sú v GloMoSime obsiahnuté smerovacie protokoly AODV, DSR, LAR, WRP, ZRP a FishEye. Z transportnej vrstvy sú dostupné TCP a UDP protokoly, na aplikačnej vrstve je možné použiť protokoly FTP, Telnet, HTTP alebo CBR. Nevýhodou je zastaralosť nástroja, malý počet používateľov GloMoSimu a z toho vyplývajúca menšia podpora. K naprogramovaniu modelu je potrebná čiastočná znalosť jazyka PARSEC. K nástroju

chýba rozsiahlejšia dokumentácia.

5.2.4 QualNet Network Simulator

Qualnet je komerčný nástroj umožňujúci simulácie pevných a bezdrôtových sietí vyvíjaný spoločnosťou Scalable Network Technologies. Nástroj funguje na platformách typu Unix, Linux, Windows a Mac OS X. QualNet vznikol zo simulačného nástroja GloMoSim. Základ QualNetu tvorí simulačné jadro, ktoré obsahuje základnú funkcionalitu potrebnú pre simuláciu diskrétnych udalostí. Nad týmto jadrom je vrstva skladajúca sa z knižníc, obsahujúcich modely, ktorých priebeh je možné simulovať, napr. model reprezentujúci satelitnú sieť, senzorovú sieť alebo model reprezentujúci UMTS sieť. V týchto knižniciach je dostupná aj široká škála protokolov používaných v MANET sieťach ako aj sada protokolov používaných pre zabezpečenie Quality of Service. Ovládanie nástroja zabezpečuje vrstva rozsiahleho grafického užívateľské rozhrania QualNet Developer, ktoré sa skladá z rôznych komponent. Komponenty sú rozdelené podľa typu činnosti, pre ktorú je komponenta navrhnutá. Napríklad pre návrh simulačného scenára je možné použiť komponentu QualNet Scenario Designer, pre zobrazenie priebehu simulácie sú určené komponenty QualNet Animator a QualNet 3D Visualizer, výsledky je možné analyzovať po skončení simulácie z uložených súborov pomocou Qualnet Analyzera. Ovládanie a návrh simulácie sú možné aj pomocou príkazového riadku bez nutnosti použiť grafické rozhranie. Oproti GloMoSimu obsahuje QualNet podstatne viac modelov a protokolov, prepracovanejšie grafické rozhranie a lepšiu podporu koncových používateľov.

5.2.5 OPNET (Optimized Network Engineering Tools)

Je nástroj určený pre simuláciu, analýzu a návrh komunikačných sietí, sieťových protokolov alebo aplikácii. Vývoj začal v roku 1986 firmou MIL3, Inc., v súčasnosti sa firma nazýva OPNET Technologies, Inc. Opnet je používaný hlavne v komerčnej sfére, pre vzdelávacie účely je možné získať bezplatnú licenciu. OPNET funguje na platformách Linux a Windows. Pôvodne bol navrhnutý pre simuláciu klasických káblových sietí, ale súčasné verzie podporujú už aj bezdrôtové siete. Obsahuje veľké množstvo modelov, ktoré je možné pri simulácii použiť. Simuláciu je možné prevádzať na rôznych úrovniach detailu. Štruktúra OPNET-u ako aj návrh simulačného scenára sú rozdelené do 3 úrovní/domén:
Network domain - doména obsahujúca celkovú sieť, topológiu siete, mobilitu prvkov v sieti atď...
Node domain - obsahuje jednotlivé uzly, z ktorých sa skladá sieť (routre, switche, koncové stanice, atď...)
Process domain - obsahuje procesy a subprocesy, ktoré prebiehajú v jednotlivých uzloch siete
Návrh jednotlivých modelov sa deje pomocou grafických rozhraní. Uzly sú zložené z vrstiev podobne ako je to v modeli OSI a jednotlivé vrstvy sa skladajú z procesov. Procesy sú modelované vo forme konečného automatu, kde sú popísané stavy konečného automatu a prechodové funkcie medzi týmito stavmi. Každý stav konečného automatu je na úrovni kódu implementovaný v jazykoch C alebo C++. Návrh celkovej siete, jej topológie a správania sa deje takisto pomocou grafického rozhrania, kde užívateľ vyberie prvky, z ktorých sa má sieť skladať a navzájom ich poprepája. Vzhľadom k tomu, že je OPNET komerčný produkt,

je dodávaný s rozsiahlou používateľskou dokumentáciou a výrobca poskytuje používateľom podporu pri problémoch. Výhodou pre používateľov je prepracované grafické rozhranie.

5.2.6 NS3

Vznikol z nástroja NS2, ma ale novú, prepracovanú štruktúru, takže nie je spätne kompatibilný s NS2. Vývoj začal približne v roku 2008. NS3 je bezplatné šíriteľný, open-source software určený pre výskumné a vzdelávacie účely v oblasti sieťových technológií. Cieľové platformy, pre ktoré je zatiaľ nástroj určený sú Linux a MAC OS X, prípadne Windows s použitím Cygwin. Cieľom vývoja bol vznik ľahko rozšíriteľného simulačného nástroja, ktorého modely budú veľmi blízko odpovedať realite. Do tohto nástroja by malo byť možné jednoducho integrovať nové alebo existujúce moduly a nástroj mal v sebe zahŕňať podporu pre určitú formu virtualizácie. Jadro NS3 je napísané podobne ako NS2 v C++. Oproti NS2 obsahuje množstvo vylepšení, príkladom je použitie smart pointrov zabezpečujúcich automatickú dealokáciu nepoužívanej pamäte alebo možnosť definície callbackov pre rôzne akcie. Definícia a konfigurácia simulačných scenárov je taktiež písaná v jazyku C++, prípadne pomocou jazyka Python spojeného s C++. V NS3 nie je na rozdiel od NS2 použitý jazyk OTcl. Oproti NS2 zatiaľ NS3 obsahuje len malú časť modulov. Väčšinu modulov z NS2 je potrebné kompletne prepísať, aby bolo možné tieto moduly začleniť do NS3. Nevýhodou NS3 je malý počet modulov a taktiež zatiaľ menší počet používateľov oproti NS2. Dokumentácia NS3 je momentálne taktiež na slabšej úrovni ako u predošlej verzie NS2.

Okrem popísaných nástrojov existuje ešte množstvo ďalších, nástroje uvedené v predošlých odsekoch sú ale podľa dostupných zdrojov najrozšírenejšie a existuje k ním najviac referencií. Z výberu použiteľných nástrojov som hneď na začiatku vylúčil komerčné nástroje OPNET a QualNet z dôvodu potreby zakúpenia licencie a menšej používateľskej základne. Vývoj GloMoSimu je zastavený a nástroj je už neaktuálny, preto bol z výberu následne vylúčený tiež. Zo zostávajúcich uvedených nástrojov sa mi podľa referencií najviac pozdávali nástroje OMNeT++ a NS2. Hlavné dôvody pre výber OMNeT++ a NS2 boli open-source a veľkosť používateľskej komunity. Vyšší počet používateľov zaistí ľahšie riešenie prípadných problémov, ktoré sa mohli počas práce vyskytnúť. Ďalším z rozhodujúcich dôvodov bola podpora platformy Linux a Windows, na ktorých som chcel produkt používať. Výhodou bola aj relatívna aktuálnosť produktov, v čase keď som vyberal z dostupných produktov bola posledná vydaná verzia OMNeTu (verzia 4.0) z roku 2009. Posledná vydaná verzia NS2 (verzia 2.34) bola z roku 2008. V súčastnosti (6/2010) už je OMNeT vydaný vo verzii 4.1 a u NS2 sa pripravuje na vydanie verzia 2.35, ktorej vydanie je plánované okolo polovice roku 2010.
Na začiatku práce som si z uvedených dvoch nástrojov vybral OMNeT++, aj keď následne som prešiel k použitiu NS2. Samotný OMNeT++ neposkytuje moduly a komponenty potrebné pre simuláciu sietí. Tieto moduly a komponenty sú distribuované vo forme frameworkov. Najrozšírenejšími frameworkmi, ktoré poskytujú okrem iných modulov aj moduly pre simuláciu bezdrôtových sietí sú frameworky INETMANET [5] a MiXiM[6]. Za najviac vhodný som podľa dostupných materialov považoval INETMANET. Výhodou tohto frameworku bolo aj to, že do neho bola portovaná implementácia AODV protokolu AODV-UU z Uppsala University. Takto som mal možnosť upravovať implementáciu používanú v skutočnosti a nebolo potrebné implementovať celý smerovací protokol od začiatku.

Po oboznámení sa s prostredím OMNeTu a zvoleným frameworkom som začal s úpravou smerovacieho protokolu. Modifikácie samotného protokolu sú popísané v kapitole č. 4. Po začiatočných úpravách protokolu som začal s testovaním výkonnosti modifikovaného protokolu oproti pôvodnej implementáciou portovanou do OMNeTu. Časy, ako dlho bežali jednotlivé testovacie scenáre sa mi ale zdali značne dlhé, preto som sa rozhodol vyskúšať rovnaký simulačný scenár v nástroji NS2. Do nástroja NS2 je portovaná rovnaká verzia AODV-UU ako je v INETMANETE. V tomto nástroji boli časy behov jednotlivých simulačných scenárov kratšie, preto som sa nakoniec rozhodol pre simuláciu používať nástroj NS2, aj keď architektúra OMNeTu a konfigurácia simulačného scenára pomocou NED súborov sa mi zdali prehľadnejšie ako architektúra NS2 s použitým OTcl jazykom. V NS2 sa mi na druhej strane pozdával viac spôsob výstupu z behu simulácie, ktorý bol vo formáte textového súboru, v ktorom boli uložené udalosti predstavujúce zasielané a prijímané správy.

5.3 Existujúce implementácie AODV protokolu

Existujúce implementácie smerovacích protokolov pre MANET siete sa dajú rozdeliť do dvoch kategórií. Prvou kategóriou sú implementácie, v ktorých je smerovací protokol spustený v užívateľskom režime (user-space) podobne ako iné užívateľské aplikácie. Druhým typom je implementácia, kde je smerovací protokol spustený v rámci jadra operačného systému (kernel-space). V užívateľskom režime beží smerovací protokol ako užívateľský proces. Tento proces odchytáva odchádzajúce a prichádzajúce pakety a na základe týchto paketov zaisťuje potrebné smerovacie informácie. Výhodou protokolov implementovaných do užívateľského režimu je ich portabilita. Implementácia nie je priamo spojená s jadrom operačného systému, čo umožňuje použitie na inom zariadení alebo prípadne inom type operačného systému. Ďalšou z výhod je zložitosť inštalácie protokolu v užívateľskom režime oproti protokolu inštalovaného do jadra systému. Užívateľ si môže nainštalovať relatívne jednoducho protokol bežiaci na aplikačnej úrovni. V prípade, že by si chcel do zariadenia doplniť protokol bežiaci v režime jadra operačného systému, vyžadovalo by to úpravu a kompiláciu tohto jadra, čo nie je pre bežného užívateľa triviálna vec. Výhodou je aj rozdiel v zložitosti vývoja protokolov v jednotlivých režimoch. V prípade protokolu bežiaceho v jadre musí mať vývojár znalosť tohto jadra, náročnejšie je taktiež testovanie a ladenie behu protokolu.
Nevýhodou užívateľského režimu oproti kernel režimu je nižší výkon protokolu. Pakety a smerovacie informácie sú predávané cez viaceré rozhrania kým sa dostanú na aplikačnú úroveň, čo zvyšuje počet operácií a vytvára určité zdržanie.
V nasledujúcej časti je stručne uvedených niekoľko najdostupnejších implementácií AODV protokolu. Jedná sa o implementácie, ktoré je možné použiť v prostredí linuxových operačných systémov. Z týchto implementácii som vybral jednu, z ktorej som pri úpravách vychádzal a na ktorej prebiehalo testovanie.

Mad-hoc AODV
Táto implementácia funguje celá v užívateľskom režime, bola to prvá voľne dostupná implementácia AODV protokolu. Proces smerovacieho protokolu sleduje odchádzajúce a prichádzajúce pakety, na základe informácií získaných zo sledovaných paketov iniciuje akcie potrebné

pre zabezpečenie smerovania. Táto implementácia neobsahovala podporu pre multicast a neodpovedala plnému zneniu RFC 3561. Implementácia obsahovala viacero chýb a obmedzení, kvôli ktorým prestala byť používaná.

AODV-UU

Je vyvinutá na Uppsala University vo Švedsku. Táto implementácia funguje z časti v užívateľskom režime. Smerovací proces registruje do kernelu funkcie - tzv. callbacky, ktoré sú volané pri vybraných udalostiach, napr. pri prijatí alebo odosielaní paketu. Takto získa smerovací proces kontrolu nad posielanými paketmi a môže ich ďalej spracovávať. Implementácia odpovedá zneniu RFC, naviac obsahuje dodatočné vylepšenia AODV protokolu, ktoré je možné voliteľne zapnúť alebo vypnúť. Implementácia obsahuje napríklad mód, v ktorom uzol funguje ako internet gateway alebo rozšírenia, ktoré ošetrujú existenciu jednosmerných spojení medzi susednými uzlami. Ďalším doplnkom je voliteľný počet prijatých hello správ, ktoré musí uzol za sebou prijať, aby označil vysielajúci uzol za suseda. Týmto uzol predíde zavedeniu susedných uzlov so zlou kvalitou signálu do smerovacej tabuľky, v prípade že uzol od suseda prijal len jeden náhodný hello paket a ďalšie už nie. Výhodou implementácie AODV-UU je, že bola portovaná do simulačných prostredí OMNeT++ a NS2, takže je možné v týchto nástrojoch simulovať fungovanie protokolu, ktorý sa používa v reálnom prostredí.

AODV-UCSB

Imlementácia pre verziu kernelu 2.4, funguje podobne ako AODV-UU v užívateľskom režime, kde je implementovaná celá smerovacia logika. Všetky pakety vyžadujúce smerovaciu logiku sú predávané z kernelu smerovaciemu procesu. Nad rámec RFC implementácia dopĺňa voliteľný počet potrebných prijatých hello správ, aby bol vysielajúci uzol označený za suseda.

AODV-UIUC

Implementácia vyvinutá na University of Illinois. Je podobná implementáciam AODV-UU a AODV-USCB. Rozdiel oproti pôvodným uvedeným je ten, že v implementácii je oddelená smerovacia logika a spracovanie dátových paketov, ktoré majú byť len preposlané ďalšiemu uzlu bez potreby ďalších smerovacích operácií. Ak uzol má informáciu, kam má prijatý paket ďalej preposlať, paket sa spracuje a prepošle v režime jadra. Ak na základe prijatého paketu musí byť vykonaná operácie smerovacieho protokolu (poslanie route request alebo route error správy, paket je predaný smerovaciemu agentovi bežiacemu v užívateľskom režime. Ideou spracovania paketov, ktoré uzol len prepošle ďalej, bolo zrýchlenie spracovania paketu a obmedzenie predávania paketu z kernel-space režimu do user-space a následne po spracovaní naspäť z user-space do kernel-space režimu.

Kernel-AODV

Implementácia, kde smerovací proces beží v prostredí jadra. Výhodou je, že sa posielané a prijímané pakety nemusia predávať smerovaciemu procesu do užívateľského režimu, čo zvyšuje výkon systému. Implementácia obsahuje tiež dodatočné vylepšenia oproti RFC 3561, napríklad podporu internet gateway, podporu pre viac sieťových rozhraní alebo podporu pre multicastové vysielanie. V procese bolo obsiahnutá aj podpora pre monitorovanie sily signálu

prijatých správ, ak to podporoval hardware na ktorom bol smerovací proces spustený.

Podľa [1] sú odporúčanými implementáciami pre reálne použitie verzie Kernel-AODV a AODV-UU, ostatné z uvedených sú buď zastaralé alebo neodpovedajú zneniu RFC 3561. V [14] je skúmaný rozdiel v rýchlosti prenosu paketov obidvoch typov implementácií, pričom v práci je uvedený aj rozdiel v dobe, za ktorú uzol zistí prerušenie spojenia so susedným uzlom pri použití Hello správ a pri použití odozvy linkovej vrstvy. Použitie odozvy z nižšej vrstvy sa ukázalo byť podstatne rýchlejšie ako použitie Hello správ.

5.4 Popis testovacieho prostredia

Cieľom tejto práce bolo preskúmať vplyv metriky použitej v AODV protokole na výkon a chod celej siete a navrhnúť vhodné alternatívne metriky, ktoré by bolo možné použiť s AODV protokolom miesto pôvodnej metriky. Súčasťou tohto návrhu mala byť aj implementácia daných metrík, aby bolo možné porovnať výkon navrhnutých metrík oproti štandardnej metrike použitej v AODV protokole. V kapitole 4 bolo ukázané, že nie vo všetkých prípadoch prípadne všetkých typoch sietí je metrika založená na počte uzlov danej trasy najvhodnejším kritériom, podľa ktorého je možné vyberať trasy v sieti. Môžu existovať aj iné metriky, založené na odlišných kritériach ako je hopcount, ktorých použitím je možné zlepšiť funkčnosť siete podľa zvoleného parametru. Ako testovaciu verziu protokolu, do ktorého som implementoval vybrané metriky, bola použitá implementáciu AODV-UU z Uppsala University. Výhodou tejto implementácie je, že je portovaná do simulačných nástrojov OMNeT++ a NS2, takže je možné modifikovať, testovať a porovnávať implementáciu smerovacieho protokolu, ktorá je používaná v reálnom prostredí. Verzia AODV-UU použitá na testovanie v nástroji NS2 bola 0.9.5, pri testovaní bola použitá verzia NS2 2.34.

Ako testovacie prostredie bola použitá štvorcová oblasť s rozmermi 550 x 550 metrov. V testovanej sieti sa v jednotlivých scenároch nachádzalo postupne 10, 20, 30, 40 a 50 uzlov. Uzly sa pohybovali v prostredí náhodným pohybom. Pred začatím pohybu uzol náhodne vygeneroval súradnice miesta z použitej plochy, ktoré bolo určené ako cieľ, kam sa bude uzol pohybovať. Takisto náhodne bola vygenerovaná rýchlosť pohybu uzla - touto rýchlosťou sa uzol konštantným pohybom presúval k cieľu. Pri generovaní rýchlosti bola maximálna možná rýchlosť v jednotlivých scenároch postupne 2, 5, 8, 11 a 14 m/s. Po dosiahnutí cieľa uzol ostal na dosiahnutom mieste náhodný čas zvolený z intervalu 1 až 20 sekúnd a následne opakoval popísaný postup. Pre pohyb uzlov boli použité dva modely pohybu, a to *Random Waypoint model* a *Gauss-Markovov model*. Scenáre pohybu jednotlivých uzlov boli vygenerované použitím programov *setdest* zahrnutého v distribúcii NS2 a BonnMotion [3].
Ako model propagácie rádiového signálu bol použitý Ricianov model z [2]. Uzly v sieti boli pri použití tohto modelu a nastavenej hodnote sily vysielaného signálu schopné prijať vo vzdialenosti 140 metrov od vysielača približne 80% vyslaných paketov, vo vzdialenosti 180 metrov to bolo približne 40% vyslaných paketov. V iných prácach je často použitý model šírenia signálu Two-ray ground model a dosah uzlov nastavený až na 250 metrov. Toto nastavenie podľa môjho názoru neodpovedá reálnemu prostrediu, väčšina komunikácie v takomto prostredí prebieha priamo medzi zdrojovým a cieľovým uzlom bez potreby použitia medziuzlov.

Pri takejto priamej komunikácii sa neprejavia výhody alebo nevýhody smerovacieho protokolu s danou metrikou, preto bol v mojom testovacom scenári nastavený dosah uzlov tak, ako je uvedené v predchádzajúcom texte. Scenár komunikácie medzi jednotlivými uzlami bol vygenerovaný pomocou upraveného skriptu *cbrgen.tcl*, ktorého pôvodná verzia je zahrnutá v distribúcii NS2. V sieti bola pre vysielanie dát vybraná vždy približne polovica všetkých uzlov. Uzol vybraný pre komunikáciu si náhodne zvolil cieľový uzol, čas začiatku komunikácie a dobu z intervalu 15 až 25 sekúnd, po ktorú vysielal k cieľovému uzlu pakety s frekvenciou 4 paketov/sekundu. Každý vyslaný paket obsahoval 512 bytov dát. Po uplynutí zvolenej doby mal uzol vo vysielaní pauzu 20 sekúnd a opakoval vysielanie k novému náhodne zvolenému uzlu podľa uvedeného postupu. Každý testovací scenár bol prevedený 5 krát a z výsledných hodnôt bol spočítaný priemer. Pre každý uvedený počet uzlov bola prevedená simulácia s jednotlivými uvedenými rýchlosťami, takže spolu bolo vytvorených 5 x 5 simulačných scenárov. Pre spracovanie výsledkov boli použité mnou vytvorené skripty v nástrojoch *bash* a *awk*, pre tvorbu grafov bol použitý program *gnuplot*.

Ako hodnotiace kritéria výkonu jednotlivých metrík boli použité nasledujúce ukazovatele:

Úspešnosť doručenia vyslaných správ (Packet Delivery Ratio)
Úspešnosť doručenia vyslaných správ je pomer počtu doručených dátových správ k počtu všetkých vyslaných dátových správ. Dátovými správami označujem všetky správy vytvorené vyššou vrstvou ako sieťovou. Packet delivery ratio ukazuje kvalitu trás v sieti, označuje koľko percent vyslaných správ sa po ceste stratilo a koľko z vyslaných správ bolo úspešne doručených. V ideálnom prípade sa hodnota tohto ukazovateľa blíži k 100%, kedy sú všetky vyslané správy úspešne doručené adresátovi.

Oneskorenie (Delay)
Oneskorenie definuje zdržanie, ktoré vznikne počas prenosu, to je doba medzi vyslaním správy a jej prijatím cieľovým uzlom. V tejto hodnote je zahrnutá

- doba, ktorú je správa spracovávaná pred odoslaním u odosielajúceho uzla
- doba, ktorú správa čaká vo vstupných a výstupných frontách správ jednotlivých uzlov nachádzajúcich sa na ceste medzi odosielateľom a príjemcom
- doba, ktorú trvá samotný prenos správy po jednotlivých spojoch medzi uzlami na ceste medzi odosielateľom a príjemcom

Snahou smerovacích protokolov je udržať hodnoty delay na čo najnižších úrovniach. Pre prenosy dát s real-time charakterom je hodnota delay jedným z kľúčových ukazovateľov.

Réžijné náklady (Overhead)
Kritérium režijných nákladov popisuje mieru operácii potrebných pre chod smerovacieho protokolu. V tejto práci používam normalizovaný overhead, ktorý je definovaný ako počet vyslaných a preposlaných správ smerovacieho protokolu na prijatie jednej správy dátového charakteru. Čím je tento pomer nižší, tým nižšie sú režijné nároky protokolu a tým efektív-

nejšie smerovací protokol pracuje.

Počet zaslaných Route Error správ
Toto kritérium ukazuje stabilitu a kvalitu vybraných trás v sieti. Ukazovateľ značí počet výpadkov na existujúcich spojeniach, ktoré je potrebné opraviť. Snahou protokolu by malo byť minimalizovať počet týchto chýb a prerušení existujúcich spojení.

Kapitola 6

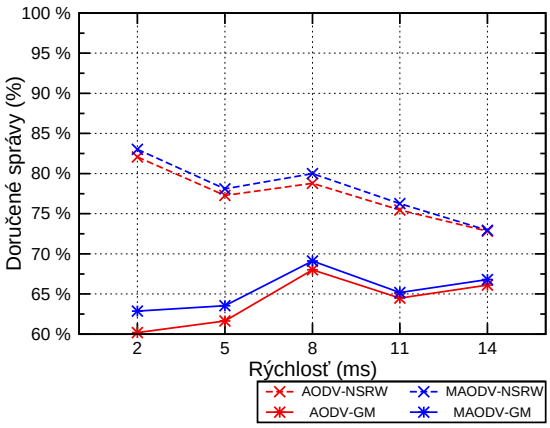
Výsledky a porovnanie jednotlivých modifikácií

6.1 MAODV

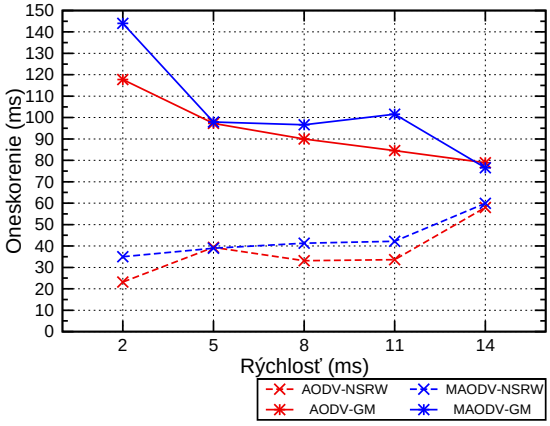
Protokol s názvom MAODV vyberá trasy, ktoré majú minimálny súčet stupňov uzlov, z ktorých sa trasa skladá. Stupňom uzla je definovaný počet uzlov v dosahu vysielania daného uzla.

Cieľom tejto metriky bolo minimalizovať počet uzlov nachádzajúcich sa v pásme rušenia, ktoré vznikne prenosom správ po vybranej trase. Očakávané výsledky s použitím tejto metriky boli zníženie hodnoty oneskorenia a zvýšenie úspešnosti doručenia vyslaných správ. U väčšiny kritérií boli výsledky, ktoré táto metrika v uvedených testovacích scenároch produkovala, približne na úrovni hodnôt výsledkov produkovaných algoritmom s pôvodnou metrikou. Očakávané zlepšenie u sledovaných kritérií sa nepotvrdilo, v prípade určitého zlepšenia to bolo na relatívne nízkej úrovni.

Obr. 6.1: PDR vs rýchlosť (30 uzlov)



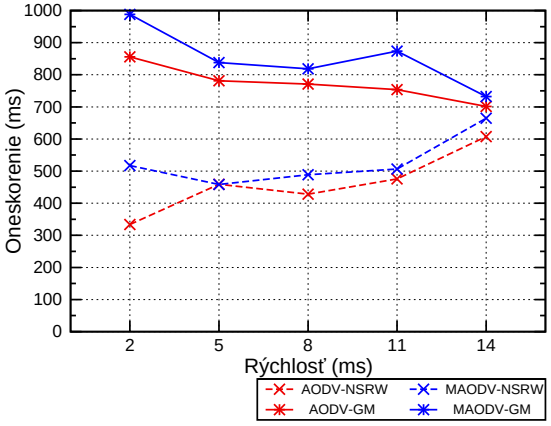
Obr. 6.2: Oneskorenie vs rýchlosť (30 uzlov)



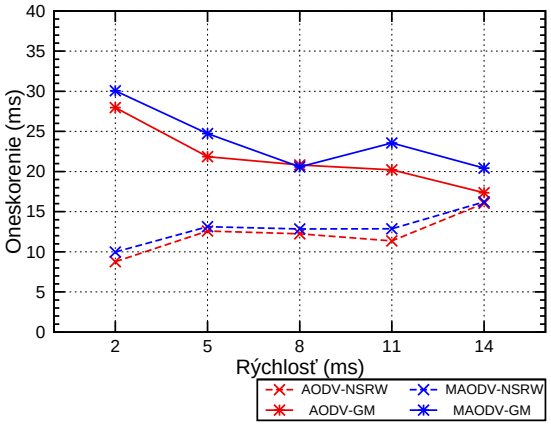
U kritéria úspešnosti doručenia vyslaných paketov nastalo mierne zlepšenie v scenároch so strednou hustotou uzlov približne od 20 do 40 uzlov. V scenári s 10 a 50 uzlami boli dosiahnuté hodnoty približne rovnaké v obidvoch verziách protokolu. Žiaden z dosiahnutých rozdielov nebol veľmi výrazný, zlepšenie sa pohybovalo v rozmedzí približne 1%-2%. So zvyšujúcou rýchlosťou uzlov je možné pozorovať klesajúcu tendenciu úspešnosti doru-

čenia správ. Dôvodom je vyššia pravdepodobnosť prerušenia používanej trasy.

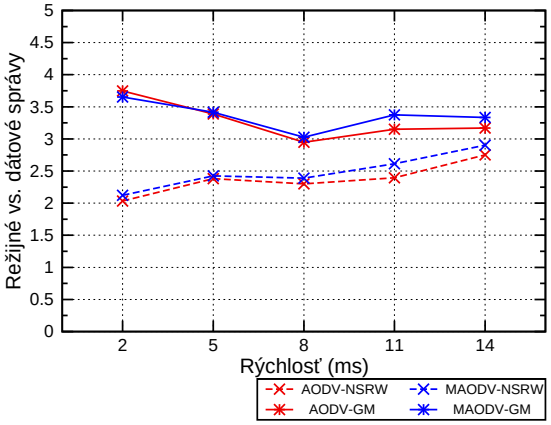
Obr. 6.3: Vyhľadanie trasy vs rýchlosť (30 uzlov)



Obr. 6.4: Prenosové oneskorenie (30 uzlov)



Obr. 6.5: Réžia vs rýchlosť (30 uzlov)



Pri prenosoch s použitím modifikovanej verzie protokolu vzniká väčšie oneskorenie ako u pôvodnej verzie protokolu. Až na ojedinelé výnimky bolo zvýšenie v rozsahu 10%-30%, čo v testovacích scenároch predstavuje zvýšenie o približne 10 milisekúnd. U sietí s menším počtom uzlov vznikli rozdiely väčšieho rozsahu, u hustejšie obsadených sietí boli rozdiely menšie, čo mohlo byť zapríčinené vznikom izolovaných uzlov alebo skupín uzlov v redšie obsadených sieťach. Z grafov sa dá pozorovať očakávaný nárast oneskorenia s rastúcou rýchlosťou pohybu uzlov v sieti.

Z grafov 6.3 a 6.4 vidno, že vzniknuté rozdiely spôsobuje hlavne dlhšia doba, za ktorú zdrojový uzol nájde cestu k cieľovému uzlu. U modifikovanej verzie je táto doba vyššia kvôli času, po ktorý majú uzly po prijatí Route Request správy túto správu uloženú v buffri pred ďalším spracovaním. Samotný čas prenosu po nájdení trasy je u modifikovanej verzie vyšší len o jednotky milisekúnd, podstatne výraznejšie zdržanie predstavuje hlavne doba medzi vyslaním Route Request správy a prijatím odpovede na túto správu.

Modifikovaná verzia protokolu potrebuje k fungovaniu smerovacieho protokolu mierne väčšie množstvo kontrolných správ ako je to u pôvodnej verzie. Celkový nárast počtu vyslaných kontrolných správ na jeden doručený dátový paket sa pohyboval v rozmedzí 10%-20%. Pri pohybe podľa Gauss-Markovho modelu bolo vyslaných viac kontrolných správ ako u Random-Waypoint modelu dôsledkom vyššej dynamiky pohybu a kratšej životnosti vytvorených trás.

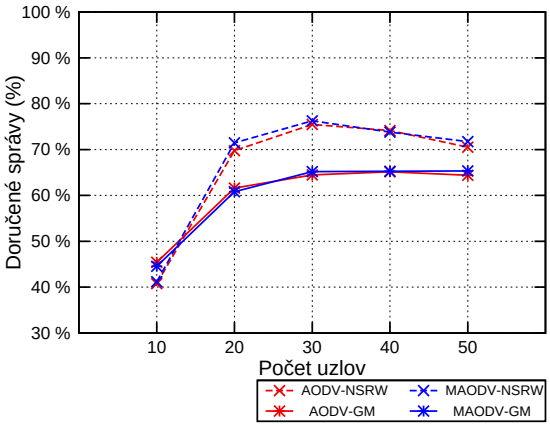
Z grafov zobrazených na obrázkoch 6.6 až 6.9 je možné pozorovať závislosť sledovaných kritérií na hustote uzlov v sieti.

Úrovne úspešnosti doručenia vyslaných správ sú najnižšie v riedko obsadených sieťach kde sú vzdialenosti medzi uzlami relatívne veľké. S nárastom počtu uzlov stúpa aj úspešnosť doručenia, najvyššia je v sieti s 30 uzlami. Od tejto hustoty má úspešnosť so zvyšujúcim počtom uzlov u Random-Waypoint modelu mierne klesajúci charakter. Príčinou je pravdepodobne rušenie, ktoré vzniká vysielaním okolitých uzlov a s rastúcou vzdialenosťou znižuje možnosť korektne prijať vysielané správy.

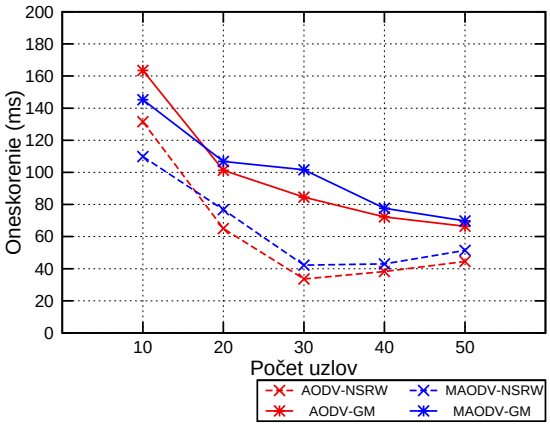
Rovnaký stúpajúci charakter s rastúcim počtom uzlov je možné pozorovať aj pri pomere kontrolných správ na jednu doručenú dátovú správu a pri počte vyslaných Route Error správ. Oproti pôvodnej verzii protokolu sa znížil počet správ typu Route Error približne o 10%.

Doba oneskorenia prenosu klesá s rastúcou hustotou uzlov v sieti. V hustejšie obsadených sieťach majú uzly viac susedov, s čím rastie pravdepodobnosť skoršieho prijatia odpovede na vyslanú Route Request správu.

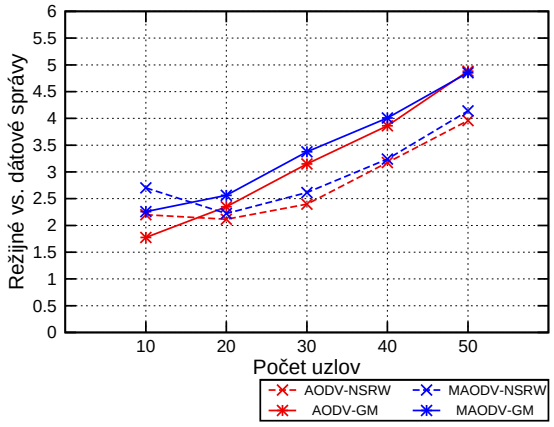
Obr. 6.6: PDR vs počet uzlov (11 m/s)



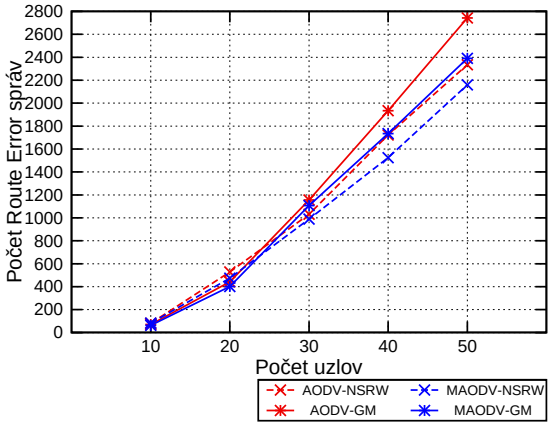
Obr. 6.7: Oneskorenie vs počet uzlov (11 m/s)



Obr. 6.8: Réžia vs počet uzlov (11 m/s)



Obr. 6.9: Route Error správy vs počet uzlov (11 m/s)



6.2 DMAODV

Protokol s názvom DMAODV používa metriku s obráteným ohodnotením ako je to u verzie MAODV. Táto modifikácia preferuje trasy s maximálnym súčtom stupňov uzlov, z ktorých sa trasa skladá.

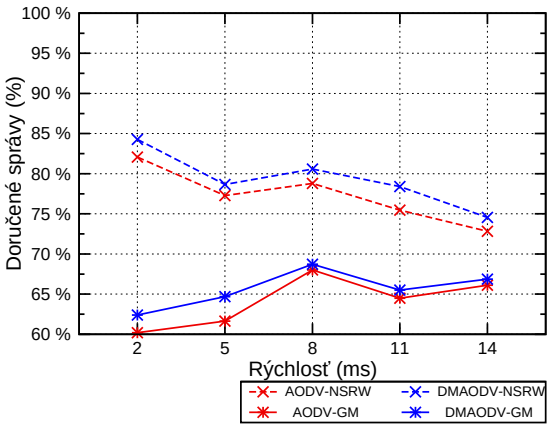
Podkladom k návrhu tejto metriky bol predpoklad, že chyby ktoré vznikajú na trase je možné opraviť rýchlejšie, ak sa v okolí prerušeného spoja nachádza vyšší počet uzlov. Cieľom tejto metriky bolo zvýšiť úspešnosť doručenia vyslaných správ. Výsledky simulácii ukázali, že použitím tejto metriky sú hodnoty sledovaných kritérií na lepšej úrovni oproti pôvodnej verzii ako i verzii MAODV.

Úspešnosť doručenia vyslaných správ bola pri obidvoch modeloch pohybu vyššia ako u pôvodnej verzie, v prípade Random-Waypoint modelu bolo toto zlepšenie na vyššej úrovni oproti Gauss-Markovmu modelu. Zlepšenie nastalo približne o 1-2 %, čo nie je nijako výrazné.

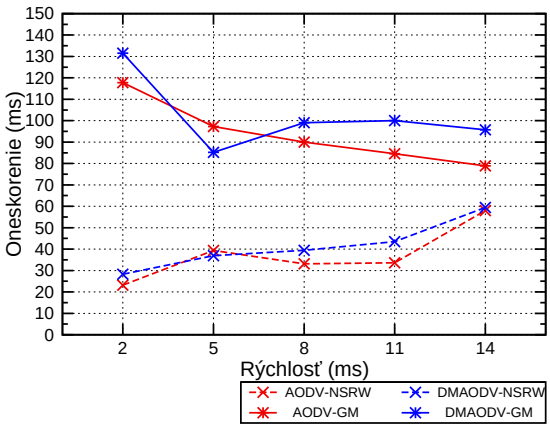
Hodnoty oneskorenia boli podobne ako u verzie MAODV vyššie ako hodnoty dosiahnuté s originálnou verziou. Dôvodom je buffrovanie prijatých Route Request správ po určitú dobu pred následným spracovaním. Zvýšenie oproti originálnej verzii je približne v rozsahu 10 až 20%, čo v prípade siete s 30 uzlami a pohybom podľa Gauss-Markovho modelu predstavuje oneskorenie vyššie približne o 10 milisekúnd.

U tejto verzie bol pozorovateľný výraznejší pokles počtu Route Error správ, kde bol tento počet nižší oproti originálnej verzii približne o tretinu. Tento pokles je zapríčinený možnosťou opraviť prerušenie trasy lokálne, kedy nie je potrebné zasielať Route Error správu na základe ktorej sa iniciuje nové vyhľadanie trasy.

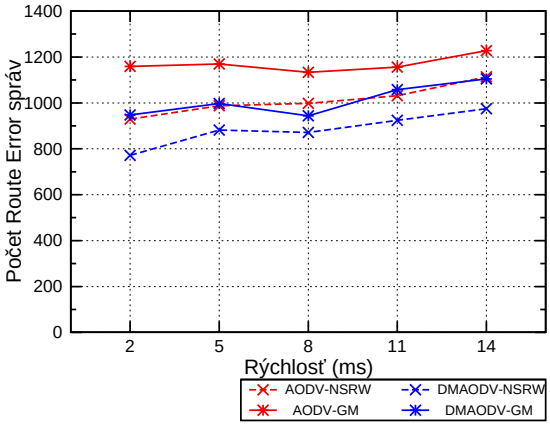
Obr. 6.10: PDR vs rýchlosť (30 uzlov)



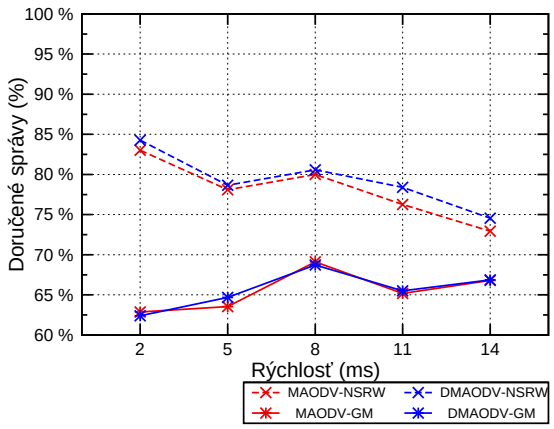
Obr. 6.11: Oneskorenie vs ryclosť (30 uzlov)



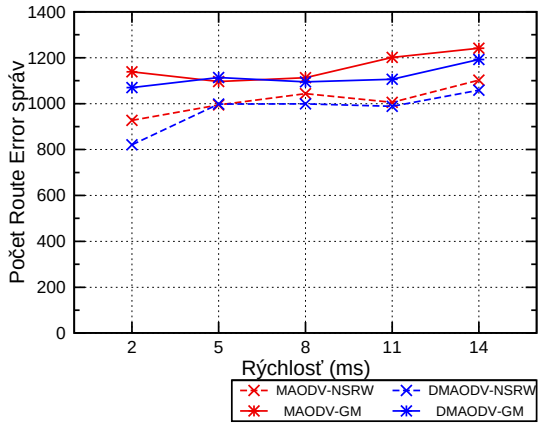
Obr. 6.12: Route Error správy vs rýchlosť (30 uzlov)



Obr. 6.13: PDR vs rýchlosť (30 uzlov)



Obr. 6.14: Route Error správy vs rýchlosť (30 uzlov)

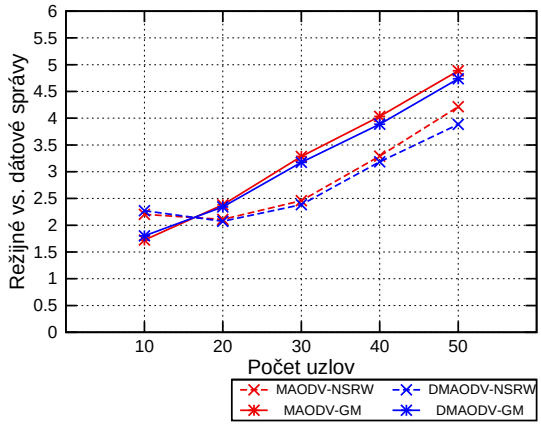


Z grafov na obrázkoch 6.13 až 6.16 vidno rozdiely pri porovnaní verzie MAODV a verzie DMAODV.
Úspešnosť doručenia vyslaných správ je mierne vyššia u verzie DMAODV. Pri pohybe uzlov podľa Random-Waypoint modelu je možné toto zvýšenie pozorovať zreteľnejšie, pri druhom pohybe sú hodnoty na približne rovnakej úrovni. S klesajúcou rýchlosťou klesá úspešnosť obidvoch verzií protokolu približne rovnakým tempom.

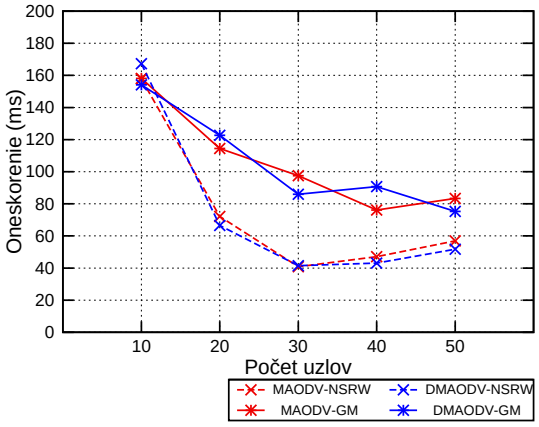
Pri počte generovaných Router Error správ vidno pokles počtu týchto správ u verzie DMAODV, čo je spôsobené vyšším počtom susedných uzlov.

Oneskorenie pri prenose správ je u obidvoch protokolov na približne rovnakých hodnotách, až na mierne výkyvy pri Gauss-Markovom modeli. Verzia DMAODV používa pre potreby smerovacieho procesu menší počet kontrolných správ ako verzia MAODV. U scenárov s menším počtom uzlov 10 a 20 je počet týchto kontrolných správ vyšší ako u pôvodného protokolu, u hustejšie obsadených sietí je ale tento pomer obrátený v prospech verzie DMAODV.

Obr. 6.15: Réžia vs počet uzlov (11 m/s)



Obr. 6.16: Oneskorenie vs počet uzlov (11 m/s)



6.3 TRMAODV

TRMAODV je verzia založená podobne ako MAODV na rušení v okolí uzlov trasy. Táto verzia vyberá trasy podľa objemu dát, ktoré sú vysielané jednotlivými uzlami trasy, ich susednými uzlami a na základe počtu uzlov, z ktorých sa trasa skladá.

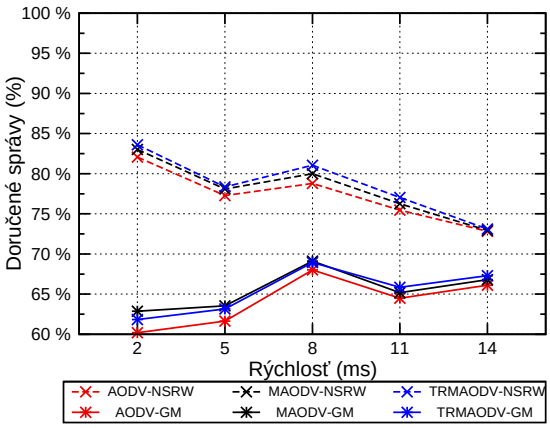
Protokol sa snaží vybrať z dostupných trás tú trasu, ktorá sa skladá z uzlov, v okolí ktorých a cez ktoré je vysielaný čo najmenší objem dát. Týmto by sa malo dosiahnuť rozloženie zaťaženia medzi rôzne uzly v sieti.

Úspešnosť doručených správ sa pohybuje približne 2% nad hodnotami dosiahnutými použitím originálneho protokolu. Táto úroveň je mierne vyššia aj ako bola úroveň dosiahnutá pri verzii MAODV. Viditeľnejšie zlepšenie nastalo pri scenároch s 20 a 30 uzlami. Pri scenároch s ostatnými počtami uzlov bolo zlepšenie v menšom rozsahu ako pri predchádzajúcich uvedených scenároch.

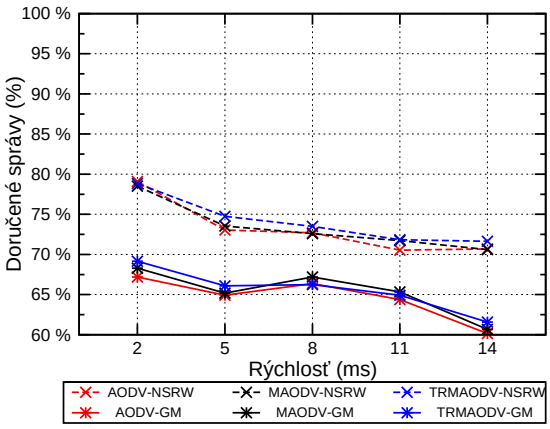
Z grafu na obrázku 6.19 je vidieť priemerný počet použitých kontrolných správ na doručenie dátovej správy v scenároch pri pohybe uzlov s rýchlosťou 8 m/s. Počet týchto správ je nepatrne vyšší ako u originálnej verzie protokolu, zvýšenie je skoro rovnakého rozsahu ako u verzie MAODV. Z grafu je taktiež pozorovateľný nárast použitého počtu kontrolných správ na doručenú dátovú správu s rastúcou hustotou uzlov v sieti. U sietí s menším počtom uzlov je oproti originálnej verzii výraznejší nárast použitých kontrolných správ, u sietí s vyšším počtom uzlov (30 a viac) je tento pomer približne na rovnakej úrovni ako v pôvodnej verzii.

Pomer kontrolných správ na doručenú dátovú správu sa zdá byť vysoký, ale tento pomer je závislý na frekvencii, s akou uzly vysielajú dátové správy. Ak by všetky uzly vysielali do siete viac správ, bol by tento pomer podstatne nižší.

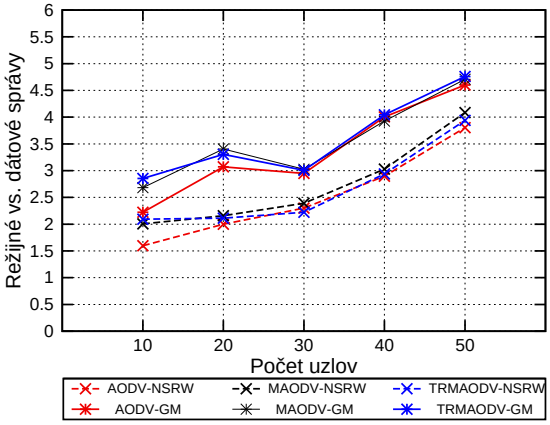
Obr. 6.17: PDR vs rýchlosť (30 uzlov)



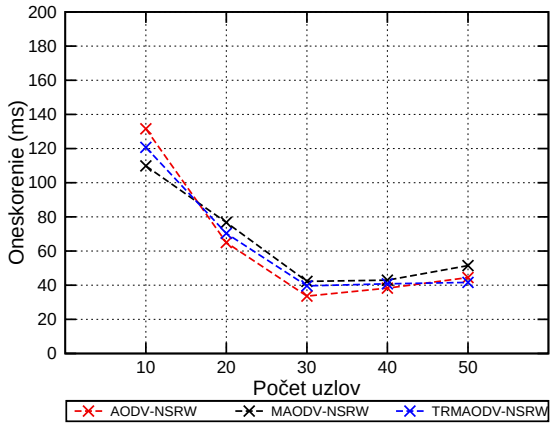
Obr. 6.18: PDR vs rýchlosť (50 uzlov)



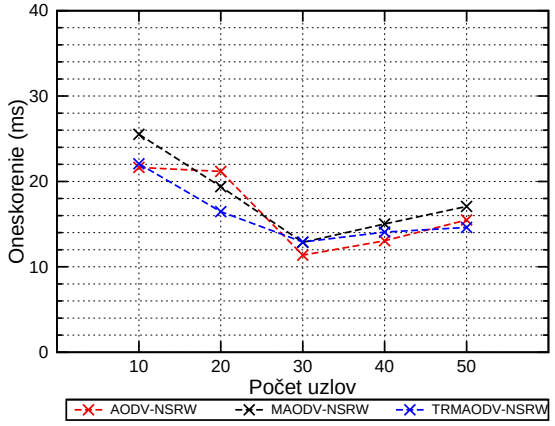
Obr. 6.19: Réžia vs počet uzlov (8 m/s)



Obr. 6.20: Oneskorenie vs počet uzlov (11 m/s)



Obr. 6.21: Prenosové oneskorenie vs počet uzlov (11 m/s)

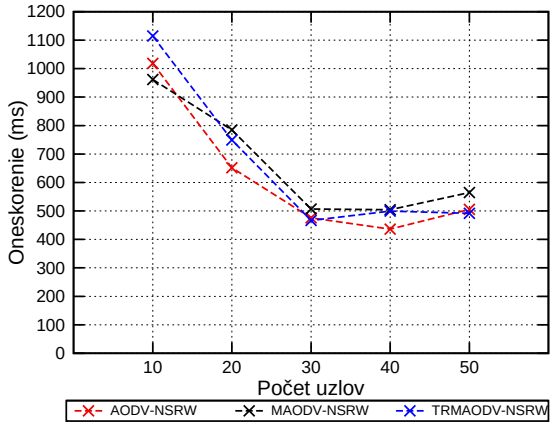


Na obrázku 6.20 je znázornené oneskorenie, ktoré vzniká pri prenose v nasimulovaných scenároch s počtami uzlov od 10 do 50 s rýchlosťou pohybu 11 m/s. Z grafu vidno, že oneskorenie vznikajúce pri použití tejto modifikácie je v rozmedzí medzi oneskorením vzniknutým pri originálnej verzii a MAODV verzii. V niektorých prípadoch sú hodnoty lepšie ako obidve porovnávané verzie. Rovnaký charakter je možné sledovať u prenosového oneskorenia, ktoré je počítané od doby vyslania dátovej správy zdrojovým uzlom po doručenie tejto správy cieľovému uzlu.

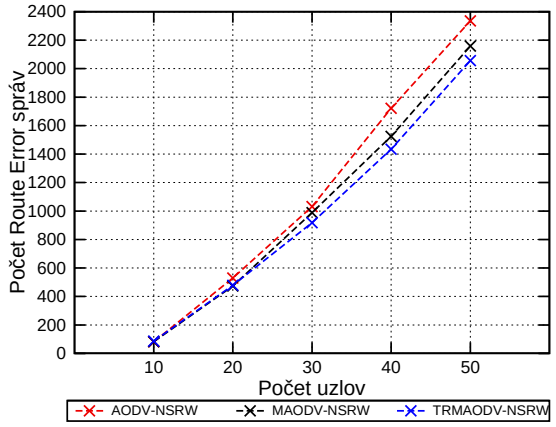
Z uvedených grafov je možné vidieť závislosť oneskorenia vzhľadom k počtu uzlov nachádzajúcich sa v sieti. U siete s 10 uzlami má oneskorenie najvyššiu hodnotu kvôli malému počtu uzlov v sieti a nízkemu stupňu uzlov. So stúpajúcim počtom prvkov rastie stupeň uzlov, uzly majú viac smerovacích informácií a pravdepodobnosť, že niektorý blízky uzol bude môcť zaslať odpoveď je vyššia.

U kritéria počtu zaslaných Route Error správ bol tento počet nižší vo všetkých testovacích scenároch približne v rozpätí 10 až 20 % oproti počtu Route Error správ zaslaných v originálnej verzii protokolu.

Obr. 6.22: Vyhľadanie trasy vs počet uzlov (11 m/s)



Obr. 6.23: Route Error správy vs počet uzlov (11 m/s)



6.4 EMAODV

Verzia s názvom EMAODV používa pre ohodnotenie trás metriku ETX. Protokol s touto metrikou vyberá trasy, na ktorých očakáva najmenší počet vyslaní správ na linkovej vrstve potrebných k úspešnému prenosu jednej správy od zdrojového uzla k cieľovému.

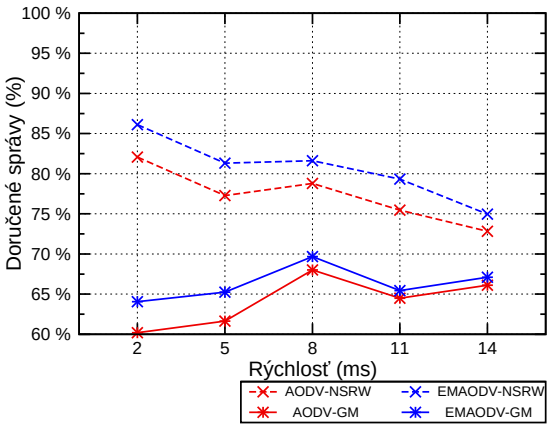
Metrika bola navrhnutá za účelom zvýšenia úspešnosti doručenia vyslaných správ. Simulácia ukázala, že použitím tejto metriky je možné dosiahnuť prenosy správ s vyššou úspešnosťou doručenia ako pomocou metriky minimalizujúcej dĺžku trasy. Negatívnou vlastnosťou je vyšší počet kontrolných správ, ktoré protokol používa.

Úspešnosť doručenia bola u obidvoch modelov pohybu s použitím ETX na vyšších úrovniach oproti originálnej verzii. V najlepších prípadoch nastalo zlepšenie približne v rozsahu 5%. Hustota uzlov v sieti nemala vplyv na rozdiel dosiahnutých hodnôt, u všetkých simulovaných počtov uzlov nastalo zlepšenie medzi pôvodným a modifikovaným protokolom približne rovnakého rozsahu.

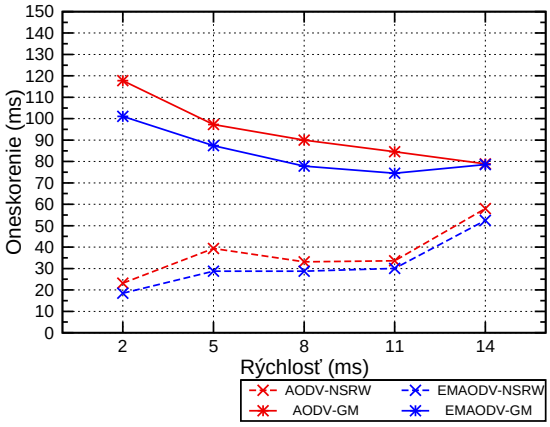
Doba oneskorenia, ktorá vznikla pri prenose v scenároch s menším počtom uzlov, bola približne rovnako dlhá u obidvoch verzií porovnávaných protokolov. U scenárov s vyššími počtami uzlov 30 a viac bola táto doba výrazne nižšia, v niektorých prípadoch to bolo zníženie až o 50 %. Toto zníženie bolo zapríčinené čiastočne aj opakovaným vyslaním Route Request správ u uzla ktorý vyslanie tejto správy inicioval.

Pri počte zaslaných Route Error správ bol zaznamenaný rapídny pokles u verzie s metrikou ETX. Počet týchto správ je nižší približne o 30% až 40% bez rozdielu na použitý model pohybu.

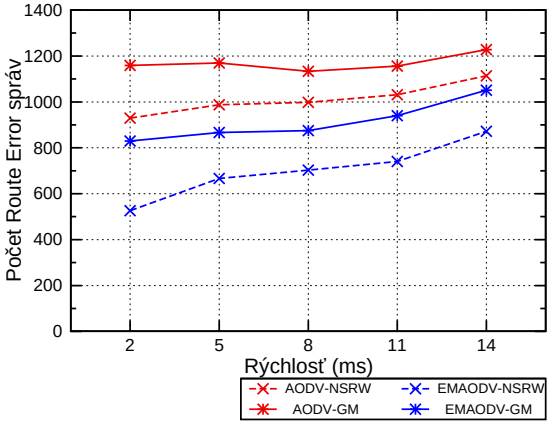
Obr. 6.24: PDR vs rýchlosť (30 uzlov)



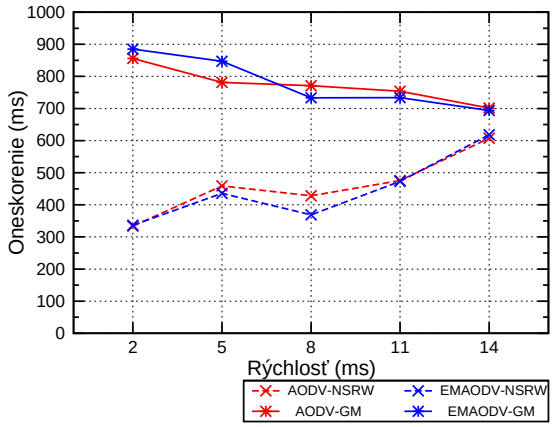
Obr. 6.25: Oneskorenie vs rýchlosť (30 uzlov)



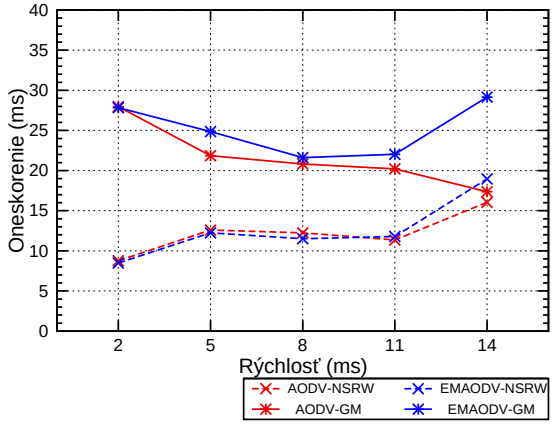
Obr. 6.26: Route Error správy vs rýchlosť (30 uzlov)



Obr. 6.27: Vyhľadanie trasy vs rýchlosť (30 uzlov)



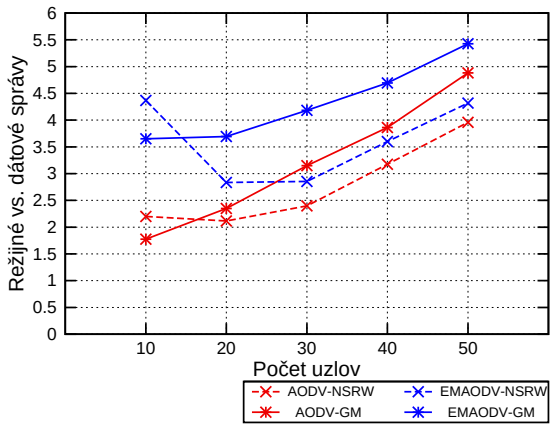
Obr. 6.28: Prenosové oneskorenie vs rýchlosť (30 uzlov)



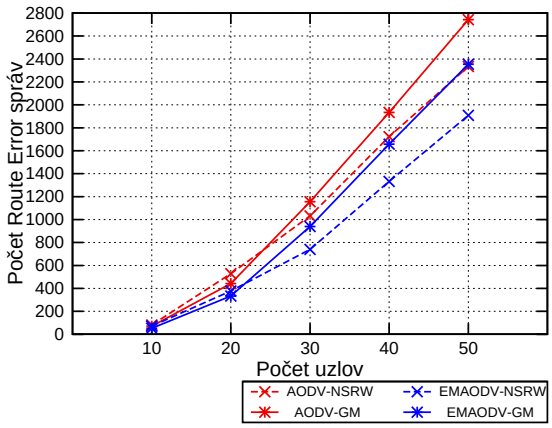
Grafy na obrázkoch 6.27 až 6.30 ukazujú porovnanie hodnôt oneskorenia v závislosti na rýchlosti a porovnanie režijných nákladov a počtu Route Error správ v závislosti na počte uzlov.
Doba oneskorenia, ktorá vzniká pri samotnom prenose správ medzi cieľovým a koncovým uzlom, je u obidvoch verzií skoro rovnaká u scenárov s počtom uzlov 20 a viac. Pri scenároch s menším počtom uzlov je táto doba u metriky ETX vyššia približne o 10 až 20 ms. U doby potrebnej pre nájdenie trasy je rozdelenie rovnaké, u scenárov s počtom uzlov 10 a 20 je doba potrebná pre vyhľadanie spojenia zhodná u obidvoch verzií, u vyšších počtov uzlov je táto doba u metriky ETX mierne nižšia.

Z obrázku 6.29 vidno nárast počtu kontrolných správ oproti nemodifikovanej verzii. Pri použití metriky ETX bol zaznamenaný najvyšší nárast počtu kontrolných správ zo všetkých verzií v tejto práci. Dôvodom tohto nárastu je pravidelné vysielanie Hello správ všetkými uzlami, tieto správy je nutné vysielať pre výpočet ETX hodnôt medzi susednými uzlami.

Obr. 6.29: Režia vs počet uzlov (11 m/s)



Obr. 6.30: Route Error správy vs počet uzlov (11 m/s)



6.5 TMAODV

Verzia protokolu nazvaná TMAODV vyberá trasy, ktoré sa skladajú zo spojov existujúcich dlhší čas ako spoje ostatných trás. Takéto trasy vyberá na základe predpokladu, že dlhšie existujúce spojenia sú stabilnejšie ako spojenia existujúce kratší čas.

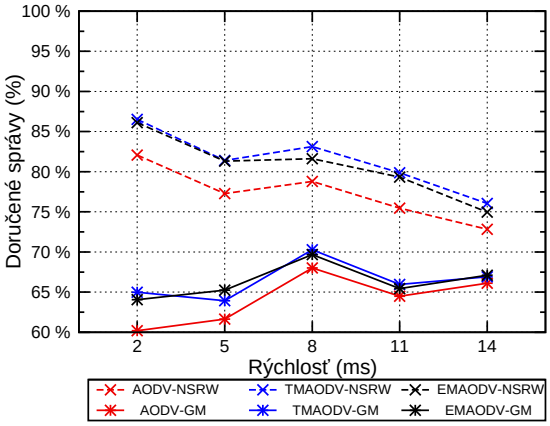
Týmto výberom sa protokol snaží predchádzať prerušeniam na zostavenej trase, dôsledkom čoho sa zvýši spoľahlivosť prenosov.

Z výsledkov simulácie scenára s 30 uzlami zobrazených na obrázku 6.31 je pozorovateľné zlepšenie v úspešnosti doručení správ približne v rozsahu, v akom to bolo pri použití metriky ETX. V niektorých prípadoch boli dosiahnuté hodnoty ešte vyššie ako s použitím ETX. Toto zlepšenie nastalo aj pri simulácii scenárov s ostatnými počtami uzlov v sieti. Rozsah zlepšenia bol u obidvoch modelov pohybu približne na rovnakej úrovni.

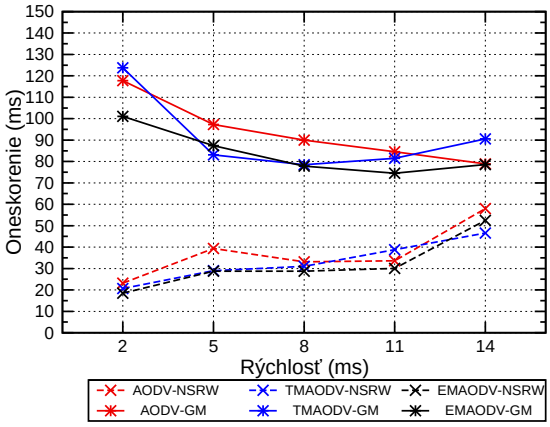
Hodnoty oneskorenia vzniknuté u verzie TMAODV boli pri Random-Waypoint pohybe približne na rovnakej úrovni ako v pôvodnej verzii, pri Gauss-Markovom pohybe boli nižšie. Výraznejšie zlepšenie oproti originálnej verzii nastalo rovnako ako pri úspešnosti doručenia až pri scenároch s viac ako 30 uzlami. Doba samotného prenosového oneskorenia bola oproti ostatným verziám rozdielna minimálne, pozorovateľnejší pokles bol zaznamenaný pri čase potrebnom na nájdenie použiteľnej trasy.

V porovnaní počtu Route Error správ vidno pokles oproti originálnej verzii protokolu približne o štvrtinu vyslaných správ, čo značí že vybrané trasy sú stabilnejšie. Oproti verzii ETX je počet týchto správ vyšší, čo je čiastočne zapríčinené aj menším množstvom Hello správ vyslaných jednotlivými uzlami.

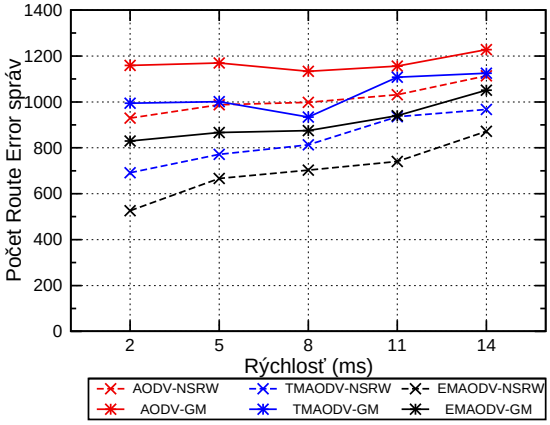
Obr. 6.31: PDR vs rýchlosť (30 uzlov)



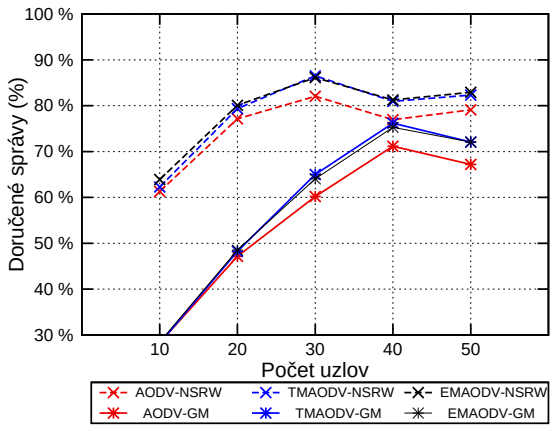
Obr. 6.32: Oneskorenie vs rýchlosť (30 uzlov)



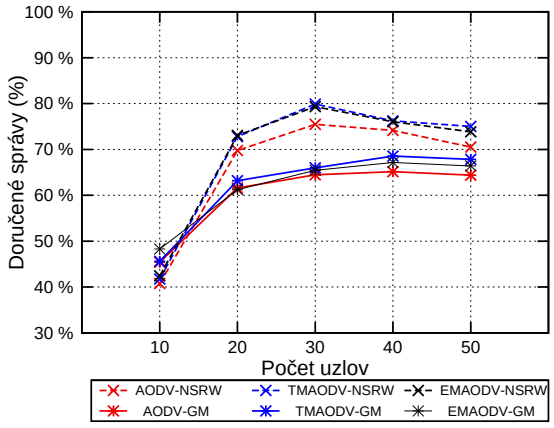
Obr. 6.33: Route Error správy vs rýchlosť (30 uzlov)



Obr. 6.34: PDR vs počet uzlov (2 m/s)



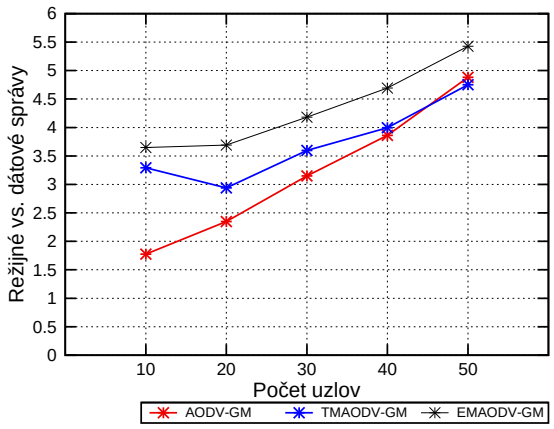
Obr. 6.35: PDR vs počet uzlov (11 m/s)



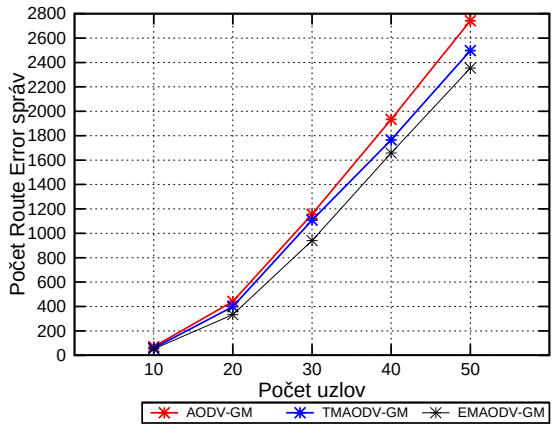
Pri porovnaní úspešnosti doručených správ v závislosti na počte uzlov v sieti je vidieť zlepšenie u sietí s 20 a viac uzlami, zlepšenie je výraznejšie s rastúcim počtom uzlov. Pri redšie obsadených sieťach je úspešnosť doručenia rovnaká ako u verzie ETX ale vyššia ako u pôvodného protokolu. Pri počte uzlov 40 a 50 je úspešnosť mierne vyššia aj oproti verzii ETX, čo je možné pozorovať z grafu na obrázkoch 6.34 a 6.35.

Pri použití verzie TMAODV je vidieť vyšší počet vyslaných kontrolných správ ako je to u originálnej verzie, ktorý je zapríčinený tým, že Hello správy vysiela každý uzol, nie len uzly nachádzajúce sa na aktívnej trase. Tieto správy uzly musia vysielať aby si udržali informácie o dĺžke spojenia so susednými uzlami. Oproti verzii používajúcej metriku ETX je počet týchto správ nižší približne o 15%.
Z grafu na obrázku 6.37 je vidieť pri rýchlosti 11 m/s pokles vyslaných Route Error správ oproti originálnej verzii približne v rozsahu 20 % . Čím viac uzlov obsahuje testovací scenár, tým viac viditeľný je rozdiel v počte vyslaných správ.

Obr. 6.36: Režia vs počet uzlov (11 m/s)



Obr. 6.37: Route Error správy vs počet uzlov (11 m/s)



6.6 SMAODV

Verzia pomenovaná SMAODV využíva k výberu trás informáciu o sile signálu a hladiny okolitého rušenia u spojov medzi jednotlivými uzlami, z ktorých sa trasa skladá.

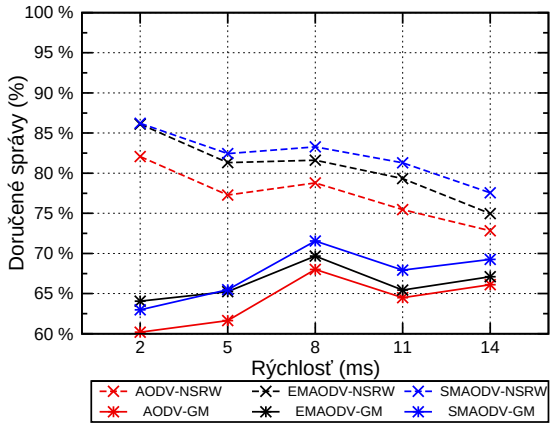
Cieľom pri návrhu metriky bolo vybrať trasu zloženú zo spojov, u ktorých je najvyššia pravdepodobnosť korektného príjmu vyslaných správ. U spojov vybraných s touto metrikou je najvyššia pravdepodobnosť príjmu správy bez chýb ešta aj v prípade, ak stúpne hladina rušenia alebo sila signálu klesne na nižšiu úroveň. Od tejto metriky bolo očakávané zvýšenie počtu doručených správ oproti počtu vyslaných správ ako aj nižší počet zaslaných Route Error správ.

U tejto metriky je zaznamenaný najvyšší nárast v úspešnosti doručených správ oproti ostatným testovaným verziám. U sietí s nízkym počtom uzlov je zlepšenie viditeľné v menšom rozsahu, s rastúcou hustotou uzlov v sieti stúpa aj rozsah zlepšenia. V scenároch s 40 a 50 nastalo zlepšenie u obidvoch modelov pohybu v rozsahu až do 5%.

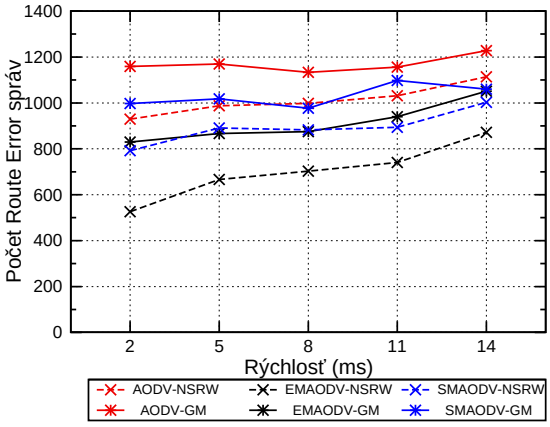
Počet zaslaných Route Error správ je nižší v rozsahu cca 10% oproti originálnemu protokolu, ale je vyšší v porovnaní s počtom Route Error správ zaslaných pri verzii EMAODV s metrikou ETX. Tento nárast je zapríčinený dôsledkom menšieho počtu Hello správ vysielaných jednotlivými uzlami.

V pomere zaslaných kontrolných správ na jednu dátovú správu táto verzia používa nižší počet kontrolných správ ako pôvodný protokol. Aj u tohto kritéria bolo zaznamenané najlepšie zlepšenie spomedzi všetkých testovaných verzií. U ostatných verzií bolo zvýšenie spôsobené vysielaním Hello správ všetkými uzlami v sieti, prípadne vyšším výskytom chýb na zostavených trasách.

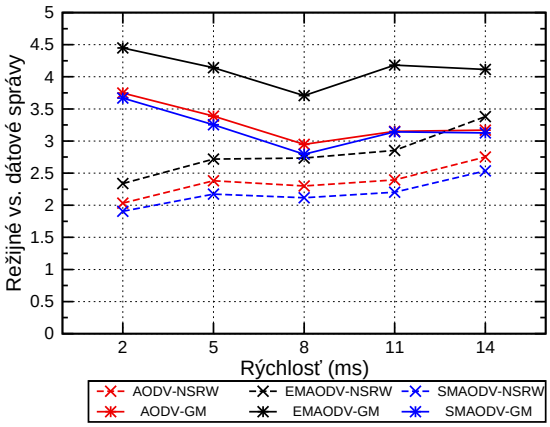
Obr. 6.38: PDR vs rýchlosť (30 uzlov)



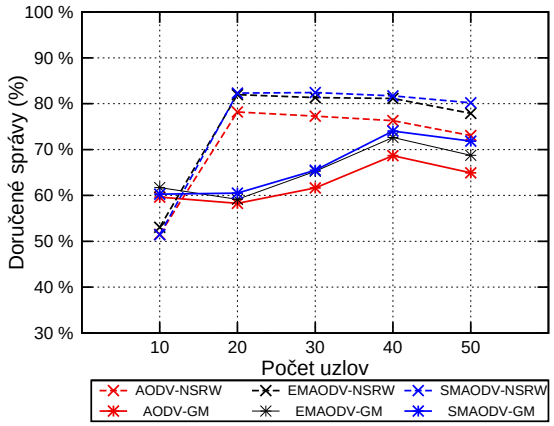
Obr. 6.39: Route Error správy vs rýchlosť (30 uzlov)



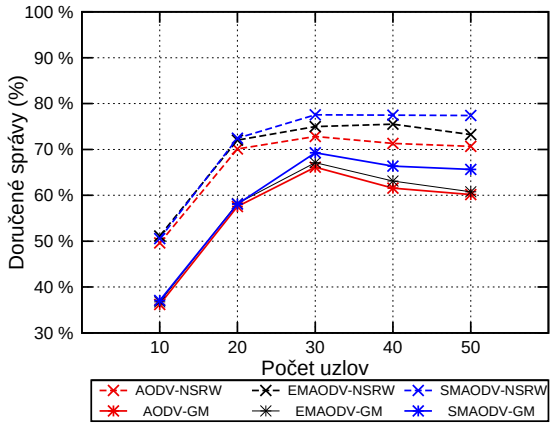
Obr. 6.40: Réžia vs rýchlosť (30 uzlov)



Obr. 6.41: PDR vs počet uzlov (5 m/s)



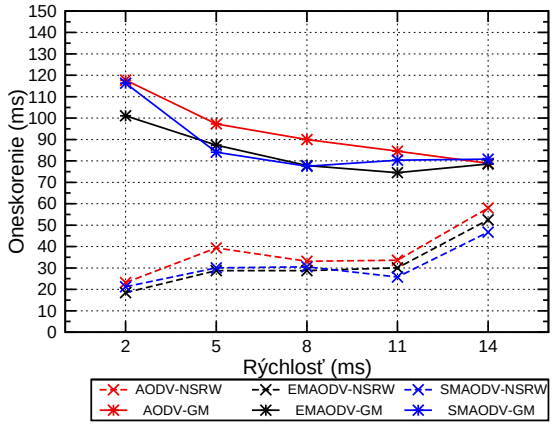
Obr. 6.42: PDR vs počet uzlov (14 m/s)



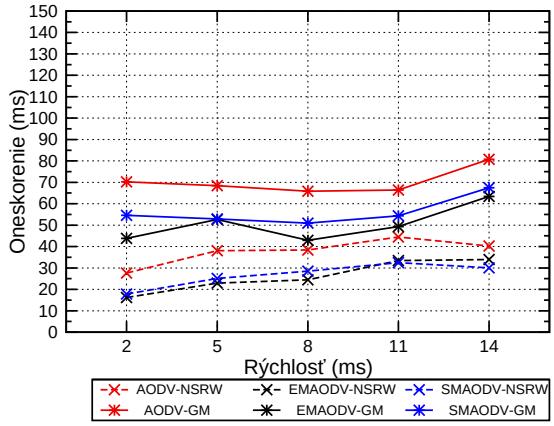
Grafy na obrázkoch 6.41 a 6.42 ukazujú porovnanie hodnôt úspešnosti doručených správ v závislosti na počte uzlov v sieti pri pohybe uzlov maximálnymi rýchlosťami 2 m/s a 14 m/s. Z grafov vidno podstatné zlepšenie verzie EMAODV oproti pôvodnému protokolu a ešte vyššie zlepšenie u verzie SMAODV. Rozdiel medzi verziami SMAODV a EMAODV je už menej výrazný ako rozdiely oproti originálnej verzii AODV, viditeľnejší je u vyššej rýchlosti uzlov. Z grafov takisto vidno, že pri riedkom obsadení siete je rozdiel medzi jednotlivými verziami na minimálnej úrovni, zvyšuje sa až so stúpajúcou hustotou uzlov v sieti.

Grafy na obrázkoch 6.43 a 6.44 zobrazujú závislosť oneskorenia a rýchlosti pohybu uzlov pri dvoch rôznych hustotách uzlov v sieti. U obidvoch scenárov je dosiahnutá nižšia doba oneskorenia pri použití verzie SMAODV. Zníženie tejto doby až na výnimky v niektorých prípadoch bolo v rozsahu okolo 20%. Z grafov sa dá pozorovať aj stúpajúca tendencia vzniknutého oneskorenia vzhľadom k maximálnej rýchlosti, akou sa uzly v sieti pohybujú.

Obr. 6.43: Oneskorenie vs rýchlosť (30 uzlov)



Obr. 6.44: Oneskorenie vs rýchlosť (50 uzlov)

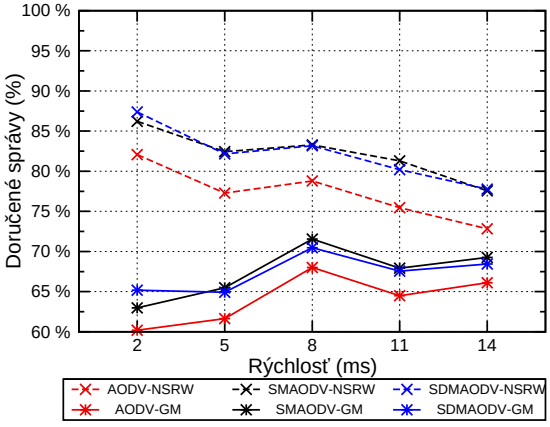


6.7 SDMAODV

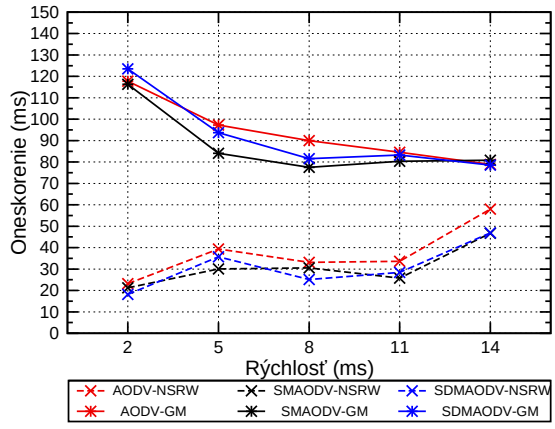
Verzia s názvom SDMAODV ohodnocuje trasy na základe kombinácie váh troch sledovaných parametrov - minimálneho odstupu signálu od rušenia vyskytujúceho sa na spojoch trasy, na základe oneskorenia, ktoré je namerané pri prenose po danej trase a na základe počtu uzlov tvoriacich trasu.

Cieľom pri návrhu tejto metriky bolo rovnako ako u predchádzajúcej verzie vybrať trasy zložené zo spojov, u ktorých je najvyššia pravdepodobnosť bezchybného príjmu správ. Vedľajším kritériom tejto metriky bolo vyberať trasy s nízkym oneskorením. U tejto metriky je možné úpravou váh pri jednotlivých zložkách ohodnotenia uprednostňovať rôzne zložky ohodnotenia. Pri uvedených výsledkoch bola hlavná zložka ohodnotenia pomer signálu a rušenia v okolí, pričom hodnoty oneskorenia a dĺžka trasy boli zahrnuté ako sekundárne ukazovatele.

Obr. 6.45: PDR vs rýchlosť (30 uzlov)



Obr. 6.46: Oneskorenie vs rýchlosť (30 uzlov)

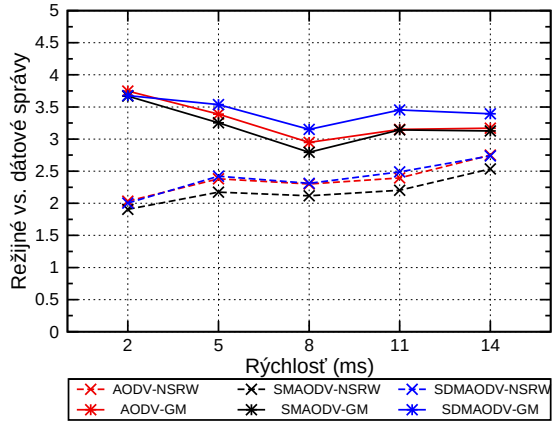


Z výsledkov získaných testovaním je možné pozorovať úspešnosť doručenia vyslaných správ na približne rovnakej alebo mierne nižšej úrovni ako u verzie SMAODV. V ojedinelých prípadoch bola úspešnosť mierne vyššia u tejto verzie, hlavne pri type pohybu s menšou dynamikou, vo väčšine prípadov je ale úspešnosť vyššia u predošlej metriky. Dosiahnuté rozdiely boli približne v rozsahu 1%.

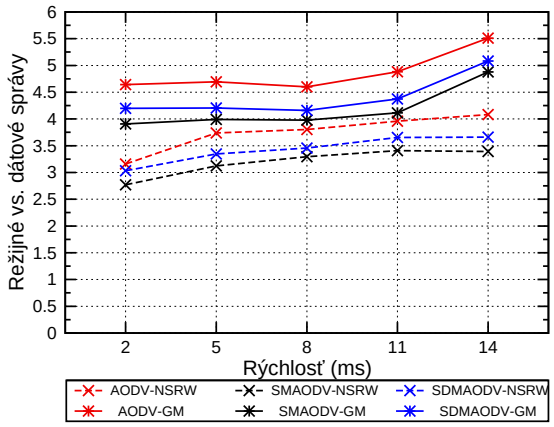
Hodnoty oneskorenia mali rovnaký priebeh aký bol u predchádzajúceho kritéria. Zlepšenie nastalo u verzie SDMAODV oproti verzii SMAODV len v ojedinelých prípadoch, častejším javom bolo mierne zvýšenie dosiahnutej doby oneskorenia.

Táto verzia vyžaduje použitie väčšieho množstva kontrolných správ. Zvýšenie je zapríčinené množstvom Hello správ, ktoré musí vysielať každý uzol v sieti, aby bolo možné pomocou týchto správ merať oneskorenie pri prenosoch medzi susednými uzlami.

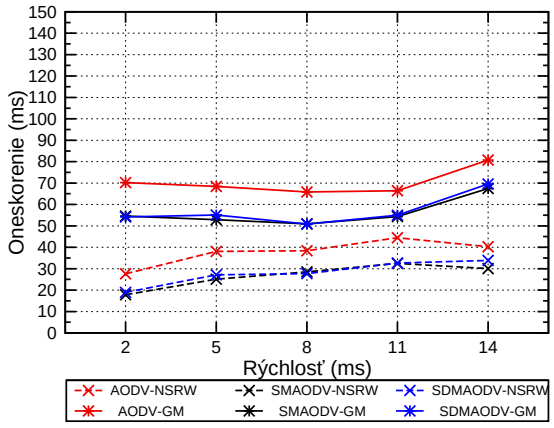
Obr. 6.47: Réžia vs rýchlosť (30 uzlov)



Obr. 6.48: Réžia vs rýchlosť (50 uzlov)



Obr. 6.49: Oneskorenie vs rýchlosť (50 uzlov)

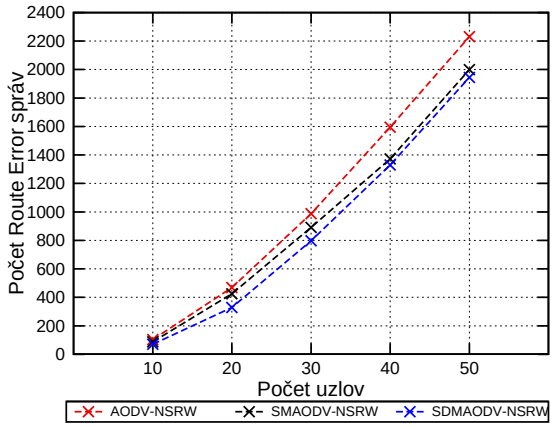


Pomer dátových a kontrolných správ bol ale v scenároch s počtom uzlov od 40 a 50 na lepších úrovniach ako u originálnej verzie protokolu. V scenároch s počtom uzlov 10 až 30 bol tento pomer vyšší vzhľadom k menšiemu počtu dátových správ. Graf 6.48 zobrazuje pomer kontrolných a dátových správ v sieti s 50 uzlami.

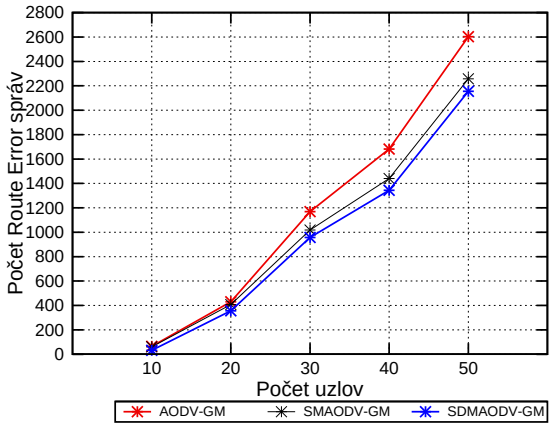
Z grafov na obrázkoch 6.50 a 6.51 sa dá pozorovať menší počet Route Error správ ako bol u pôvodnej verzie protokolu ako aj verzie SMAODV. Rozdiel je výraznejší až u hustejšie obsadených sietí, u sieti s menším počtom uzlov sú počty vyslaných Route Error správ na podobných úrovniach.

Podľa dosiahnutých výsledkov nenastalo v tejto verzii očakávané zlepšenie oproti verzii SMAODV. Nevýhodou, ktorá je cenou za počítanie oneskorenia medzi susenými uzlami, je zvýšenie počtu kontrolných správ ako aj veľkosti Hello správ, v ktorých sú hodnoty tohto oneskorenia prenášané. Klady zahrnutia prenosového oneskorenia do ohodnotenia trasy by sa mali prejaviť v sieti s nehomogénnymi uzlami, kde niektoré uzly vysielajú správy podstatne pomalšie ako ostatné uzly. V simuláčnom prostredí bolo ale možné otestovať len scenár, kde majú všetky uzly zhodne nastavené parametre.

Obr. 6.50: Route Error správy vs počet uzlov (5 m/s)



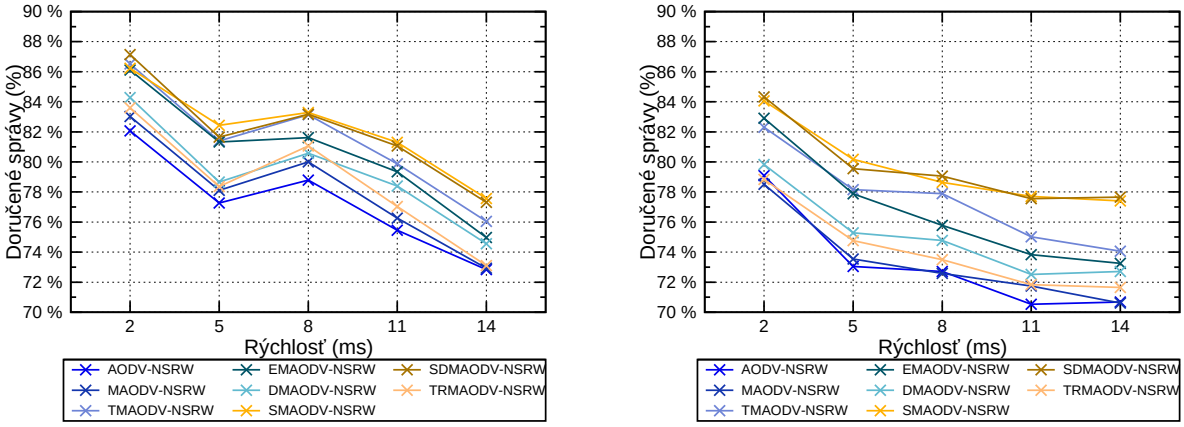
Obr. 6.51: Route Error správy vs počet uzlov (5 m/s)



6.8 Vyhodnotenie a súhrn výsledkov

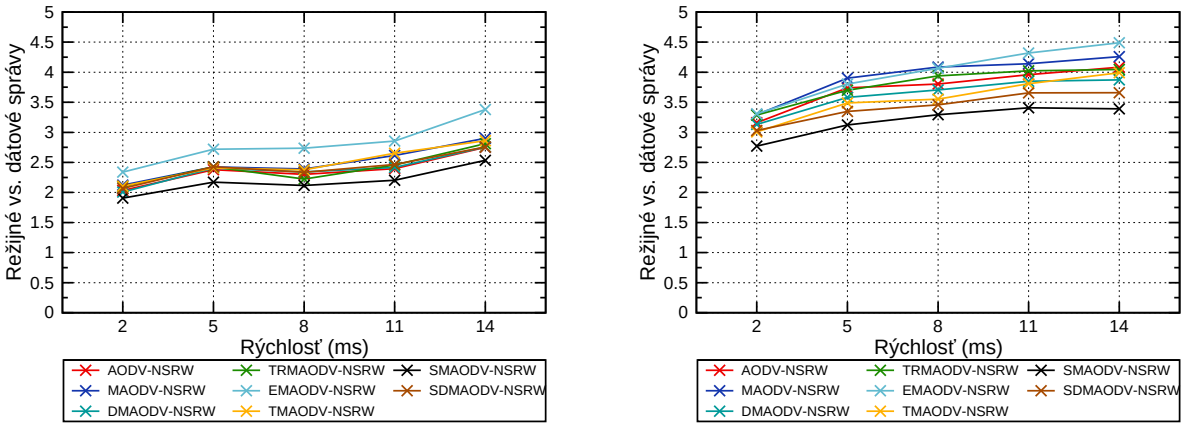
Výsledky simulačných scenárov ukázali, že zmenou kritéria, podľa ktorého sú ohodnocované a vyberané trasy je možné v niektorých prípadoch dosť výrazne ovplyvniť výkon AODV protokolu. Z prevedených simulácií je vidieť, aký vplyv majú konkrétne vybrané kritéria na chod protokolu. Niektoré verzie preukázali zlepšenie v sledovaných parametroch, u iných boli sledované parametre na rovnakých prípadne horších hodnotách, aké boli dosiahnuté použitím pôvodného protokolu.

Z pohľadu úspešnosti doručených správ je vidieť popísaný charakter verzií na grafoch znázornených na obrázku 6.52, kde je najvyššia úspešnosť dosiahnutá pri použití verzií SMAODV a SDMAODV. Ďalej v úspešnosti doručenia za týmito verziami nasledujú verzie EMAODV a TMAODV, ktoré berú do úvahy kvalitu spoja a stabilitu jednotlivých spojov. Najmenšie zlepšenie u u tohto kritéria bolo u verzií zohľadňujúcich počet susedných uzlov alebo objem prenosov v okolí vybranej trasy. U týchto metrík boli dosiahnuté hodnoty veľmi blízko hodnôt dosiahnutým pomocou originálneho protokolu.



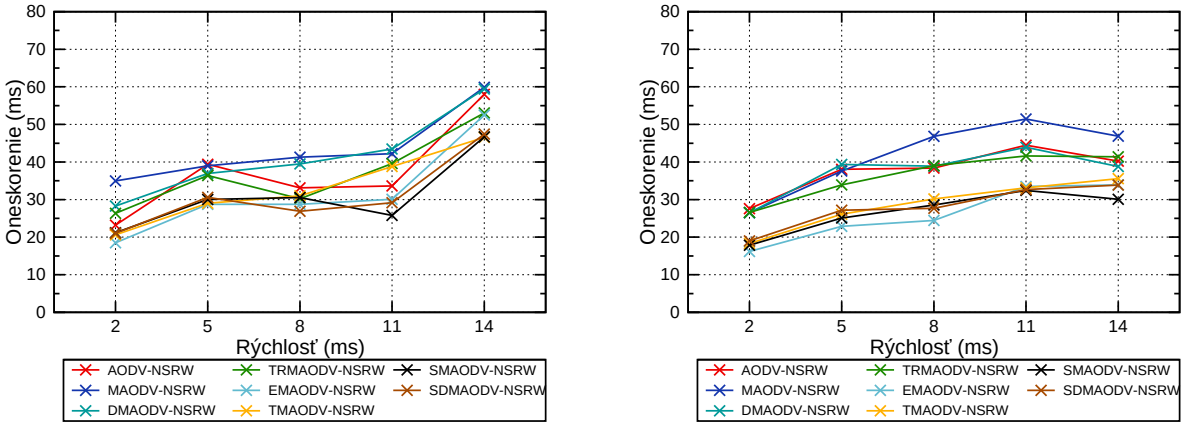
Obr. 6.52: Úspešnosť doručenia správ - 30 a 50 uzlov

Pri počte kontrolných správ v pomere na jednu doručenú dátovú správu sú u väčšiny verzií namerané hodnoty mierne vyššie ako u originálneho protokolu. Zvýšenie je približne v rozsahu 20 %. Z testovaných verzií sa výraznejšie odlišujú len verzie SMAODV, SDMAODV a TMAODV. Verzia SMAODV spotrebuje ako jediná nižší počet kontrolných správ oproti originálnej verzii aj v scenároch s menším počtom uzlov. Zvyšné dve verzie dosiahnu nižší počet kontrolných správ až v scenároch s 40 a 50 uzlami. Najviac kontrolných správ používa verzia EMAODV, hodnota u tejto verzie prevyšuje výraznejšie všetky ostatné verzie. Tento nárast vyplýva z princípu fungovania metriky ETX, ktorá potrebuje pre správny chod zvýšený počet kontrolných správ, konkrétne Hello správ. Pomer kontrolných a dátových správ nameraný v scenároch s 30 a 50 uzlami je zobrazený na obrázku 6.53.



Obr. 6.53: Režijné náklady - 30 a 50 uzlov

Z pohľadu oneskorenia je možné podľa výsledkov rozdeliť modifikované verzie na 2 skupiny - prvou je skupina verzií dosahujúca hodnoty oneskorenia približne na rovnakej a lepšej úrovni ako u originálneho protokolu, druhou skupinou sú verzie, pri ktorých bolo vzniknuté oneskorenie vyššie, prípadne rovnaké ako u originálnej verzie. Do prvej skupiny patria metriky založené na sile signálu, stabilite a kvalite spojov, čo sú SMAODV, SDMAODV, TMAODV a EMAODV. Do druhej skupiny patria zvyšné testované verzie. U týchto verzií bolo oneskorenie v scenároch s menšou hustotou uzlov na vyššej úrovni, v scenároch s vyššou hustotou bolo oneskorenie približne na rovnakej úrovni ako oneskorenie u originálneho protokolu.
Oneskorenie u všetkých modifikovaných verzií je na uvedených úrovniach aj dôsledkom dvojnásobne opakovaného vyslania Route Request správy u uzla, ktorý inicioval vyslanie tejto správy. V prípade, keď boli novovytvorené správy vysielané len jeden krát tak ako je to v originálnej verzii protokolu, oneskorenie bolo približne o 20% až 30% vyššie ako oneskorenie znázornené v grafoch.

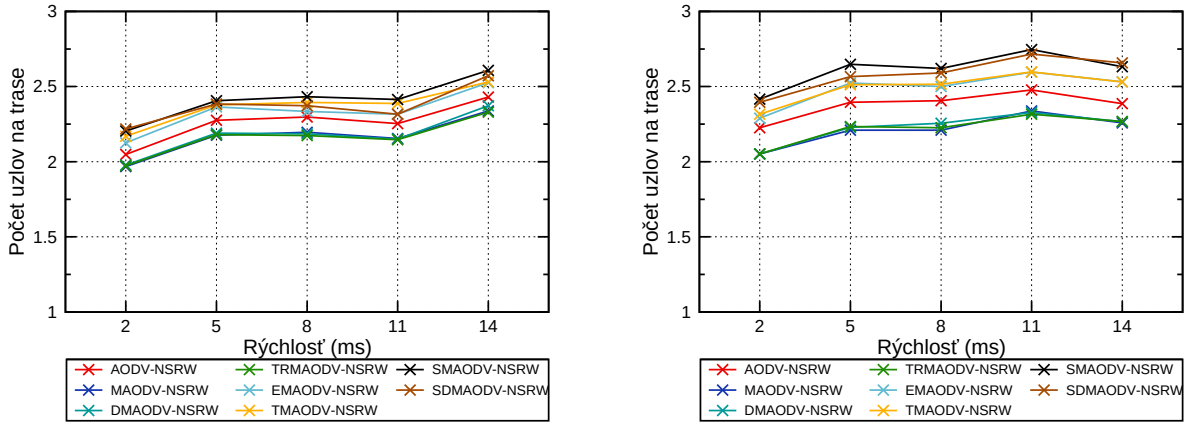


Obr. 6.54: Oneskorenie - 30 a 50 uzlov

V hodnote nameraného oneskorenia má hlavný podiel čas medzi vyslaním Route Request správy a prijatím odpovede na túto správu. Ak majú všetky uzly na trase správne smerovacie informácie, doba samotného prenosu sa medzi jednotlivými verziami líši len v jednotkách mi-

lisekúnd. Väčšie rozdiely sú v čase vyhľadania trasy, tieto doby výrazne ovplyvňujú celkovú priemernú dobu nameraného oneskorania.

Z grafov 6.54 vidno priemerné dĺžky trás, ktorými boli dátové pakety posielané. Z týchto hodnôt je možné pozorovať, že pri jednotlivých verziách boli správy posielané po rozdielnych trasách. Zo zobrazených hodnôt vidno, že napríklad u verzie SMAODV je priemerná dĺžka trasy najvyššia, ale úspešnosť aj oneskorenie sú v porovnaní s ostatnými verziami na najlepších hodnotách. Dĺžka trasy teda nie je vždy ideálnym kritériom pre výber optimálnej trasy. Naopak u verzií TMAODV a DMAODV sú dĺžky trás kratšie, ale hodnoty úspešnosti doručenia sú na nižších úrovniach oproti ostatným verziám.



Obr. 6.55: Dĺžka trasy - 30 a 50 uzlov

Celkovo je možné zhodnotiť, že najlepšie hodnoty u všetkých sledovaných kritérií boli dosiahnuté u verzií, ktoré brali do úvahy silu a kvalitu signálu, s akým boli prijímané správy na jednotlivých spojoch a stabilitu spojov trasy. U verzií, ktoré brali do úvahu počet susedných uzlov alebo objem prenosov v okolí trasy nenastalo výrazné zlepšenie, prípadne bolo len v nepatrnom rozsahu. U verzie EMAODV bola úspešnosť doručenia správ aj oneskorení pri prenose na výraznejšie lepších hodnotách oproti pôvodnému protokolu, nevýhodou ale je podstatne vyšší počet kontrolných správ potrebných pre chod protokolu.

Tabuľky s priemernými hodnotami zo všetkých meraní nameraných v jednotlivých scenároch sú uvedené v prílohe. Z kapacitných dôvodov sú v tabuľkách uvedené len výsledky scenárov s rýchlosťami 2 m/s, 5 m/s a 14 m/s, rýchlosti 8 m/s a 11 m/s sú vynechané.

Kapitola 7

Záver

V tejto práci som sa venoval smerovaciemu protokolu Ad Hoc On-Demand Distance Vector (AODV), ktorý je používaný v bezdrôtových sieťach, konkrétne v type sietí označovanom názvom Mobile Ad Hoc Networks (MANET). Tento typ sietí sa skladá z rovnocenných, mobilných a navzájom spolupracujúcich uzlov, ktoré spolu tvoria sieť. Siete MANET môžu byť vyžívané pri vojenských alebo záchranných operáciách, pre vytvorenie komunitnej siete bez potreby inštalácie pevnej infraštruktúry alebo napríklad pre zdieľanie internetového pripojenia viacerými používateľmi.
Pre tento typ sietí boli vytvorené osobitné smerovacie algoritmy rozdelené do viacerých skupín. Ja som si vybral smerovací protokol AODV, ktorý patrí do skupiny reaktívnych smerovacích protokolov. Tento protokol iniciuje vyhľadanie trasy až pri požiadavku na komunikáciu so zvoleným uzlom a trasy vyberá na základe ich dĺžky - teda počtu uzlov, ktoré danú trasu tvoria.

V práci som navrhol, skúšobne implementoval a otestoval 6 alternatívnych metrík, ktoré nahradzujú pôvodnú metriku. Cieľom práce bolo sledovať výkon protokolu s novými metrikami a porovnať namerané výsledky s výsledkami dosiahnutými pomocou pôvodného protokolu. Pri porovnávaní s originálnou metrikou som sledoval kritéria ako úspešnosť doručenia vyslaných dátových správ (paketov), oneskorenie, ktoré vzniklo počas tohto prenosu, množstvo kontrolných správ potrebných pre chod protokolu a iné. Testovanie prebiehalo v simulačnom prostredí NS2 s dvoma modelmi pohybu, maximálnymi rýchlosťami uzlov v rozsahu od 2 m/s do 14 m/s a počtom uzlov medzi 10 až 50 uzlami.

Implementované metriky je možné podľa použitých kritérií rozdeliť na dve hlavné skupiny. Prvá skupina sa snažila zlepšiť výkon protokolu minimalizovaním negatívnych vplyvov na vybrané trasy spôsobených vysielaním okolitých uzlov siete. Verzie z tejto skupiny boli pomenované MAODV a TRMAODV. Prvá verzia vyberala trasu s minimálnym počtom susedných uzlov v okolí trasy, druhá verzia vyberala trasu s minimálnym objemom vysielania v okolí trasy. Špecifickou verziou je verzia nazvaná DMAODV, ktorá vyberá trasy skladajúce sa z uzlov s najväčším počtom susedných uzlov. Ideou návrhu tejto verzie bola možnosť najrýchlejšej opravy trasy v okolí spoja, kde by vzniklo prerušenie.

Do druhej skupiny patria verzie nazvané TMAODV, SMAODV a SDMAODV. Verzie tejto druhej skupiny sa snažia vylepšiť výkon protokolu zohľadnením parametrov súvisiacich s kva-

litou signálu a stabilitou spojov na vybraných trasách do ohodnotenia trasy. Verzia TMAODV vyberá trasy skladajúce sa zo spojov existujúcich najdlhší čas, verzia SMAODV zase vyberá trasy skladajúce sa zo spojov s najkvalitnejším signálom medzi uzlami tvoriacimi spoj. Verzia SDMAODV pridáva k sile signálu ďalšie kritéria a to oneskorenie a počet uzlov na trase.
Zo získaných výsledkov popísaných v šiestej kapitole vidno, že pri zahrnutí kvalitatívnych parametrov do metriky je možné zlepšiť hodnoty sledovaných kritérií. Napríklad pri použití verzií SMAODV a SDMAODV sa podarilo zvýšiť úspešnosť doručených správ pri znížení počtu kontrolných správ a zachovaní a znížení oneskorenia pri prenose. Na druhej strane u verzií, ktoré vyberali trasu podľa počtu susedných uzlov alebo objemu prenášaného v okolí bolo zlepšenie nepatrné, tieto verzie naviac spotrebovali vyššie množstvo kontrolných paketov na jeden doručený dátový paket oproti pomeru dosiahnutému v originálnej verzii. Hodnoty oneskorenia boli u týchto verzií taktiež na vyššej úrovni oproti originálnej verzii. Počas implementácie a testovania sa mi taktiež podarilo nájsť a opraviť chybu v implementácii AODV-UU, ktorá spôsobovala v určitých prípadoch vznik smerovacích cyklov. Implementácia AODV-UU bola použitá ako základ, do ktorého som implementoval vlastné metriky.

V celkovom hodnotení s prihliadnutím na všetky sledované kritéria vykazovali metriky z druhej uvedenej skupiny lepšie výsledky ako metriky prvej skupiny. Zo získaných výsledkov vyplýva, že u Ad Hoc On-Demand protokolu alebo aj iných v praxi využívaných smerovacích protokolov používajúcich ako metriku len počet uzlov na trase, existuje potenciál na ich rozšírenie o alternatívne kritéria použité pre výber trasy, a tým aj celkové zlepšenie funkčnosti týchto protokolov.

V náväznosti na túto prácu by bolo ďalej možné otestovať chod jednotlivých verzií pri použití iných modelov prostredia, v ktorom sa nachádzajú testované uzly. V scenároch testovaných v prostredí NS2 sa napríklad nevyskytovali žiadne prekážky, zaujímavé by bolo prevedenie testov v scenári predstavujúcom prostredie zastavané budovami alebo obsahujcom iné objekty. V takomto prostredí by pravdepodobne mohli byť pomocou navrhnutých metrík dosiahnuté ešte lepšie výsledky v porovnaní s pôvodnou verziou. Taktiež by bolo zaujímavé otestovať chod verzií s inými modelmi predstavujúcimi fyzickú a linkovú vrstvu, ako sú modely dostupné v NS2, ktoré boli použité pri simulácii. Ďalšou možnosťou je prevedenie testov v prostredí obsahujúcom nehomogénne uzly, keďže v NS2 bolo možné simulovať len scenáre, v ktorých majú všetky uzly rovnako nastavené parametre vysielacieho zariadenia. Každopádne oblasť ad-hoc sietí a komunikácii v týchto sieťach poskytuje svojím rozsahom množstvo príležitostí, ktoré sú hodné za ďalší výskum.

Literatúra

[1] Ad hoc On Demand Distance Vector. http://moment.cs.ucsb.edu/AODV/.

[2] Additions to the NS network simulator to handle Ricean and Rayleigh fading. http://www.ece.cmu.edu/wireless/downloads/ns2_ricean_dist.tgz.

[3] BonnMotion - A mobility scenario generation and analysis tool. http://net.cs.uni-bonn.de/wg/cs/applications/bonnmotion.

[4] GloMoSim. http://pcl.cs.ucla.edu/projects/glomosim.

[5] INETMANET - INET with mobile/ad-hoc prototocols. http://github.com/inetmanet/inetmanet.

[6] MiXiM (Mixed simulator). http://mixim.sourceforge.net.

[7] RFC3561 - Ad hoc On-Demand Distance Vector (AODV) Routing.

[8] SIPAODV-UUETX. http://www.cs.kau.se/cs/prtp/pmwiki/pmwiki.php?n=MeshWikipage.SIPAODV-UUETX.

[9] *AODV-BR: backup routing in ad hoc networks*, volume 3, 2000.

[10] M. Ad, S. J. Dajiang, D. He, and J. Rao. A Prediction-based Link Availability Estimation for Mobile Ad Hoc Networks, 2001.

[11] W. AlMobaideen. SPDA: Stability Based Partially Disjoint AOMDV. *European Journal of Scientific Research, Vol. 27, No. 3*, pages 342–348, 2009.

[12] Aune, Frank. Cross-Layer Design Tutorial. *http://www.iet.ntnu.no/projects/cuban/archive/1812041* 2004.

[13] M. Burhkart. Analysis of Interference in Ad-Hoc Networks, 2003.

[14] I. D. Chakeres and E. M. Belding-Royer. AODV Routing Protocol Implementation Design. In *ICDCSW '04: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops - W7: EC (ICDCSW'04)*, pages 698–703. IEEE Computer Society, 2004.

[15] I. D. Chakeres and L. Klein-berndt. AODVjr, AODV simplified. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3:100–101, 2002.

[16] Z. Cheng. Exploring long lifetime routing (LLR) in ad hoc networks. In *in Proc. ACM MSWiM*, pages 203–210. ACM Press, 2004.

[17] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR), 2003.

[18] D. S. J. De, C. Daniel, A. Benjamin, A. Chambers, and R. Morris. Performance of multihop wireless networks: shortest path is not enough. *SIGCOMM Comput. Commun. Rev*, page 2003.

[19] D. S. J. De Couto. *High-throughput routing for multi-hop wireless networks*. PhD thesis, 2004. Supervisor-Morris, Robert T.

[20] M. Drini and T. Saadawi. Performance of ad-hoc routing protocols with link state. In *ICAI'09: Proceedings of the 10th WSEAS international conference on Automation & information*, pages 52–57. World Scientific and Engineering Academy and Society (WSEAS), 2009.

[21] Y. C. Fuad Alnajjar. SNR/RP AWARE ROUTING ALGORITHM: CROSS-LAYER DESIGN FOR MANETS. In *International Journal of Wireless and Mobile Networks*, pages 127–136, 2009.

[22] P. Gupta, S. Member, and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46:388–404, 2000.

[23] M. Haenggi and D. Puccinelli. Routing in Ad Hoc Networks: A Case for Long Hops. *IEEE Communications Magazine*, 43:93–101, 2005.

[24] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. pages 66–80, 2003.

[25] D. Johnson. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, 2004.

[26] C. Kim, E. Talipov, and B. Ahn. A Reverse AODV Routing Protocol in Ad Hoc Mobile Networks. 4097:522–531, 2006.

[27] Y. X. Liran Ma, Qian Zhang and W. Zhu. Interference aware metric for dense multi-hop wireless networks. In *2005 IEEE International Conference on Communications*, volume 2, pages 1261 – 1265, 2005.

[28] H. Lundgren and C. Tschudinjune. Coping with Communication Gray Zonesin IEEE 802.11b based Ad hoc Networks.

[29] T. W. M. Bandai. An On-Demand Routing Using Signal Strength for Multi-Rate Ad Hoc Networks. *IEICE TRANSACTIONS on Communications Vol.E90-B No.9*, pages 2504–2512, 2007.

[30] H. Oh. A Link Availability Predictor for Wireless Sensor Networks.

[31] G. Pei, M. Gerla, and T.-W. Chen. Fisheye State Routing in Mobile Ad Hoc Networks. In *In ICDCS Workshop on Wireless Networks and Mobile Computing*, pages 71–78, 2000.

[32] C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. pages 234–244, 1994.

[33] C. E. Perkins and E. M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *IN PROCEEDINGS OF THE 2ND IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pages 90–100, 1997.

[34] R. B. S. Waharte, B. Ishibashi. Interference-Aware Routing Metric for Improved Load Balancing in Wireless Mesh Networks. *IEEE International Conference on Communications*, pages 2979 – 2983, 2008.

[35] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37:164–194, 2005.

[36] S. Shakkottai and P. C. Karlsson. Cross-Layer Design for Wireless Networks. *IEEE Communications Magazine*, 41:74–80, 2003.

[37] a. L. Sobrinho, Jo˙Network routing with path vector protocols: theory and applications. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 49–60. ACM, 2003.

[38] C. Toh. A novel distributed routing protocol to support ad-hoc mobile computing. *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, pages 480–486, 1996.

[39] S. C. V. Park. Temporally-Ordered Routing Algorithm (TORA), 2001.

[40] X. Zhang, Q. Liu, D. Shi, Y. Liu, and X. Yu. An Average Link Interference-Aware Routing Protocol for Mobile Ad Hoc Networks. In *ICWMC 07: Proceedings of the Third International Conference on Wireless and Mobile Communications*, page 10. IEEE Computer Society, 2007.

[41] P. S. Zygmunt J. Haas, Marc R. Pearlman. The Zone Routing Protocol (ZRP) for Ad Hoc Networks, 2002.

Dodatok A

Tabuľky s výsledkami simulačných scenárov

Tabuľky s výsledkami simulačných scenárov sú uvedené na stranách 79-83:

Názvy verzií boli skrátené z dôvodu priestorových obmedzení - jednotlivé označenia sú:

M - MAODV

DM - DMAODV

TRM - TRMAODV

EM - EMAODV

TM - TMAODV

SM - SMAODV

SDM - SDMAODV

10 uzlov	AODV	M	DM	TRM	EM	TM	SM	SDM
	Random Waypoint model - max. rýchlosť 2 m/s							
PDR (%)	61.28	60.44	62.35	61.87	63.91	62.28	61.49	63.86
Oneskorenie (s)	116.422	89.721	82.535	90.780	86.981	115.903	98.007	85.934
Kontr./dát. správy	1.58	1.87	1.85	1.90	2.75	2.42	1.89	2.30
Poč. kontr. správ	1808	2050	2097	2092	3201	2745	2078	2681
Poč. HELLO správ	544	532	544	524	1804	1291	548	1309
Poč. RERR správ	116	110	112	104	82	96	111	77
	Random Waypoint model - max. rýchlosť 8 m/s							
PDR (%)	50.49	50.67	50.26	49.22	51.47	50.50	49.27	50.22
Oneskorenie (s)	126.240	159.511	137.444	110.696	102.036	103.472	78.754	101.599
Kontr./dát. správy	1.60	2.00	2.00	2.09	3.24	2.85	2.07	2.85
Poč. kontr. správ	1485	1854	1833	1850	3030	2630	1856	2585
Poč. HELLO správ	446	431	411	423	1805	1287	437	1295
Poč. RERR správ	71	77	72	81	52	58	80	51
	Random Waypoint model - max. rýchlosť 14 m/s							
PDR (%)	49.60	51.02	49.36	50.67	51.13	51.37	50.58	51.45
Oneskorenie (s)	116.414	121.104	102.092	109.105	164.580	127.424	91.359	102.044
Kontr./dát. správy	1.72	2.07	2.15	2.06	3.36	2.86	2.05	2.79
Poč. kontr. správ	1616	1955	1958	1941	3209	2714	1944	2656
Poč. HELLO správ	460	444	432	440	1804	1282	461	1298
Poč. RERR správ	102	87	92	91	80	93	95	74
	Gauss-Markov model - max. rýchlosť 2 m/s							
PDR (%)	28.27	29.23	29.30	27.75	28.46	28.21	29.54	29.65
Oneskorenie (s)	72.847	187.662	167.849	102.243	119.133	90.189	132.184	114.053
Kontr./dát. správy	2.74	3.57	3.56	3.55	6.65	5.68	3.40	5.23
Poč. kontr. správ	1348	1773	1766	1726	3321	2776	1735	2695
Poč. HELLO správ	231	227	225	226	1804	1246	238	1260
Poč. RERR správ	30	29	28	24	16	23	28	24
	Gauss-Markov model - max. rýchlosť 8 m/s							
PDR (%)	35.40	35.71	35.91	35.31	36.41	35.10	35.50	36.12
Oneskorenie (s)	82.819	133.343	139.648	90.794	126.797	105.123	107.248	118.866
Kontr./dát. správy	2.22	2.68	2.69	2.85	4.92	4.29	2.80	4.09
Poč. kontr. správ	1463	1791	1806	1838	3317	2722	1823	2662
Poč. HELLO správ	309	305	291	285	1804	1202	285	1236
Poč. RERR správ	56	49	46	50	39	40	47	45
	Gauss-Markov model - max. rýchlosť 14 m/s							
PDR (%)	36.21	37.60	38.49	36.08	36.78	36.87	37.07	37.04
Oneskorenie (s)	99.283	196.440	214.200	153.425	147.852	116.816	152.194	118.738
Kontr./dát. správy	2.55	3.06	3.00	3.25	5.49	4.63	3.19	4.57
Poč. kontr. správ	1473	1860	1860	1829	3340	2737	1839	2687
Poč. HELLO správ	307	310	318	299	1805	1247	315	1257
Poč. RERR správ	49	48	49	45	44	38	53	40

20 uzlov	AODV	M	DM	TRM	EM	TM	SM	SDM
	Random-Waypoint model - max. rýchlosť 2 m/s							
PDR (%)	77.12	78.08	76.99	77.58	80.09	79.37	80.09	79.48
Oneskorenie (s)	55.140	54.025	58.631	41.635	45.484	44.747	43.543	46.805
Kontr. vs dát. správy	1.83	1.94	1.97	1.93	2.36	2.12	1.79	2.12
Poč. kontr. správ	5383	5800	5774	5699	7191	6426	5478	6436
Poč. HELLO správ	1145	1103	1056	1111	3608	2297	1165	2310
Poč. RERR správ	475	452	439	455	290	362	411	333
	Random-Waypoint model - max. rýchlosť 8 m/s							
PDR (%)	70.39	71.75	72.55	72.39	72.43	74.03	73.90	73.03
Oneskorenie (s)	44.129	53.713	55.272	49.821	46.094	49.572	46.252	44.429
Kontr. vs dát. správy	2.00	2.16	2.05	2.11	2.75	2.29	1.97	2.34
Poč. kontr. správ	5419	5939	5733	5890	7680	6516	5603	6568
Poč. HELLO správ	1097	1005	1005	1020	3608	2243	1073	2161
Poč. RERR správ	502	456	428	441	339	399	405	368
	Random-Waypoint model - max. rýchlosť 14 m/s							
PDR (%)	70.10	70.89	70.77	69.98	72.02	72.79	72.44	73.35
Oneskorenie (s)	54.559	76.368	58.853	62.516	70.229	65.862	51.279	61.820
Kontr. vs dát. správy	2.12	2.29	2.23	2.29	2.99	2.52	2.16	2.44
Poč. kontr. správ	5649	6182	6033	6131	8175	7024	5983	6846
Poč. HELLO správ	1057	956	955	994	3609	2163	1050	2160
Poč. RERR správ	520	493	455	483	403	467	474	407
	Gauss-Markov model - max. rýchlosť 2 m/s							
PDR (%)	47.18	46.82	46.98	48.25	48.53	48.25	48.94	48.35
Oneskorenie (s)	119.962	129.110	122.336	127.145	126.416	121.421	115.652	123.501
Kontr. vs dát. správy	3.25	3.66	3.64	3.53	5.08	4.13	3.60	4.20
Poč. kontr. správ	5642	6258	6244	6266	8829	7205	6334	7331
Poč. HELLO správ	763	714	663	709	3611	1874	736	1822
Poč. RERR správ	411	349	344	351	283	322	348	324
	Gauss-Markov model - max. rýchlosť 8 m/s							
PDR (%)	49.95	49.97	50.90	51.60	50.29	52.11	52.16	51.54
Oneskorenie (s)	141.442	159.656	160.157	138.149	153.164	140.674	117.863	152.242
Kontr. vs dát. správy	3.07	3.40	3.31	3.31	4.83	3.80	3.21	3.78
Poč. kontr. správ	5675	6355	6321	6413	9092	7416	6257	7233
Poč. HELLO správ	787	709	751	740	3610	1885	793	1903
Poč. RERR správ	423	421	406	424	361	406	391	370
	Gauss-Markov model - max. rýchlosť 14 m/s							
PDR (%)	57.55	57.68	58.11	58.04	58.22	59.97	58.09	59.16
Oneskorenie (s)	118.455	133.335	127.907	114.834	105.020	128.680	126.236	110.953
Kontr. vs dát. správy	2.58	2.85	2.75	2.83	3.89	3.16	2.83	3.16
Poč. kontr. správ	5695	6341	6133	6324	8727	7340	6331	7218
Poč. HELLO správ	870	822	841	871	3611	1951	900	1958
Poč. RERR správ	494	504	480	501	421	502	497	462

30 uzlov	AODV	M	DM	TRM	EM	TM	SM	SDM
	Random-Waypoint model - max. rýchlosť 2 m/s							
PDR (%)	82.06	83.02	84.28	83.60	86.11	86.52	86.23	87.14
Oneskorenie (s)	23.194	34.992	28.268	26.332	18.504	20.617	21.184	21.030
Kontr. vs dát. správy	2.03	2.12	2.01	2.10	2.34	2.11	1.91	2.07
Poč. kontr. správ	9590	10122	9708	10065	11568	10469	9437	10342
Poč. HELLO správ	1502	1380	1464	1568	5416	3242	1700	3186
Poč. RERR správ	929	828	771	831	526	691	792	648
	Random-Waypoint model - max. rýchlosť 8 m/s							
PDR (%)	78.79	80.00	80.58	81.07	81.62	83.13	83.28	83.17
Oneskorenie (s)	33.139	41.300	39.479	30.214	28.754	31.007	30.540	26.881
Kontr. vs dát. správy	2.30	2.39	2.34	2.22	2.74	2.38	2.12	2.34
Poč. kontr. správ	10387	10948	10804	10335	12742	11315	10087	11092
Poč. HELLO správ	1484	1408	1328	1433	5414	3004	1624	3018
Poč. RERR správ	998	937	871	836	702	813	882	766
	Random-Waypoint model - max. rýchlosť 14 m/s							
PDR (%)	72.83	72.93	74.56	73.10	74.97	76.04	77.56	77.31
Oneskorenie (s)	57.973	59.902	59.409	53.051	52.483	46.561	46.705	47.430
Kontr. vs dát. správy	2.75	2.90	2.76	2.81	3.38	2.86	2.54	2.76
Poč. kontr. správ	11483	12153	11739	11803	14589	12472	11304	12273
Poč. HELLO správ	1237	1201	1172	1177	5413	2736	1377	2829
Poč. RERR správ	1113	1028	974	989	871	967	1002	908
	Gauss-Markov model - max. rýchlosť 2 m/s							
PDR (%)	60.18	62.87	62.41	61.83	64.05	64.98	62.98	65.04
Oneskorenie (s)	117.747	143.978	131.512	139.937	101.115	123.756	116.287	117.493
Kontr. vs dát. správy	3.75	3.65	3.61	3.79	4.45	3.82	3.67	3.80
Poč. kontr. správ	12294	12654	12341	12804	15658	13529	12598	13460
Poč. HELLO správ	1139	1094	1052	1053	5413	2352	1085	2339
Poč. RERR správ	1159	1003	947	988	830	994	997	913
	Gauss-Markov model - max. rýchlosť 8 m/s							
PDR (%)	68.02	69.11	68.73	68.94	69.68	70.28	71.56	70.84
Oneskorenie (s)	89.947	96.653	99.042	80.864	77.826	78.460	77.527	77.930
Kontr. vs dát. správy	2.95	3.03	2.99	3.01	3.71	3.12	2.79	3.08
Poč. kontr. správ	11458	11973	11794	11809	14772	12564	11471	12492
Poč. HELLO správ	1244	1117	1139	1227	5416	2645	1286	2659
Poč. RERR správ	1133	982	943	989	874	933	977	911
	Gauss-Markov model - max. rýchlosť 14 m/s							
PDR (%)	66.12	66.79	66.87	67.30	67.11	66.92	69.27	69.28
Oneskorenie (s)	78.897	76.526	95.730	80.726	78.547	90.533	80.869	71.255
Kontr. vs dát. správy	3.17	3.33	3.37	3.28	4.11	3.53	3.13	3.40
Poč. kontr. správ	11998	12733	12901	12676	15826	13520	12376	13461
Poč. HELLO správ	1234	1076	1024	1140	5414	2403	1223	2520
Poč. RERR správ	1227	1159	1104	1110	1051	1125	1060	1074

40 uzlov	AODV	M	DM	TRM	EM	TM	SM	SDM
	Random-Waypoint model - max. rýchlosť 2 m/s							
PDR (%)	76.95	78.08	78.28	78.04	81.23	80.93	82.65	82.57
Oneskorenie (s)	34.043	39.663	32.950	34.551	21.836	29.041	23.925	25.532
Kontr. vs dát. správy	2.83	2.89	2.78	2.93	3.11	2.86	2.50	2.77
Poč. kontr. správ	16876	17443	16825	17692	19472	17929	16034	17690
Poč. HELLO správ	1511	1483	1376	1489	7218	3360	1775	3321
Poč. RERR správ	1487	1315	1172	1341	1024	1259	1298	1237
	Random-Waypoint model - max. rýchlosť 8 m/s							
PDR (%)	76.54	77.29	77.63	76.90	78.88	79.18	81.40	81.52
Oneskorenie (s)	29.580	38.439	36.182	31.563	22.531	27.448	27.257	25.858
Kontr. vs dát. správy	2.90	3.03	2.92	2.94	3.26	2.90	2.60	2.75
Poč. kontr. správ	17057	18051	17477	17470	19817	17700	16379	17302
Poč. HELLO správ	1564	1463	1474	1465	7215	3369	1756	3350
Poč. RERR správ	1522	1426	1308	1389	1131	1301	1390	1221
	Random-Waypoint model - max. rýchlosť 14 m/s							
PDR (%)	71.32	72.84	72.31	72.48	75.49	75.29	77.48	76.44
Oneskorenie (s)	50.545	48.361	42.548	49.241	32.302	41.829	33.376	37.022
Kontr. vs dát. správy	3.32	3.39	3.37	3.45	3.69	3.39	2.99	3.22
Poč. kontr. správ	18219	19075	18805	19254	21486	19697	17860	19092
Poč. HELLO správ	1432	1332	1366	1296	7216	2947	1635	3023
Poč. RERR správ	1701	1619	1525	1563	1350	1577	1587	1475
	Gauss-Markov model - max. rýchlosť 2 m/s							
PDR (%)	76.95	78.08	78.28	78.04	81.23	80.93	82.65	82.57
Oneskorenie (s)	34.043	39.663	32.950	34.551	21.836	29.041	23.925	25.532
Kontr. vs dát. správy	2.83	2.89	2.78	2.93	3.11	2.86	2.50	2.77
Poč. kontr. správ	16876	17443	16825	17692	19472	17929	16034	17690
Poč. HELLO správ	1511	1483	1376	1489	7218	3360	1775	3321
Poč. RERR správ	1487	1315	1172	1341	1024	1259	1298	1237
	Gauss-Markov model - max. rýchlosť 8 m/s							
PDR (%)	76.54	77.29	77.63	76.90	78.88	79.18	81.40	81.52
Oneskorenie (s)	29.580	38.439	36.182	31.563	22.531	27.448	27.257	25.858
Kontr. vs dát. správy	2.90	3.03	2.92	2.94	3.26	2.90	2.60	2.75
Poč. kontr. správ	17057	18051	17477	17470	19817	17700	16379	17302
Poč. HELLO správ	1564	1463	1474	1465	7215	3369	1756	3350
Poč. RERR správ	1522	1426	1308	1389	1131	1301	1390	1221
	Gauss-Markov model - max. rýchlosť 14 m/s							
PDR (%)	71.32	72.84	72.31	72.48	75.49	75.29	77.48	76.44
Oneskorenie (s)	50.545	48.361	42.548	49.241	32.302	41.829	33.376	37.022
Kontr. vs dát. správy	3.32	3.39	3.37	3.45	3.69	3.39	2.99	3.22
Poč. kontr. správ	18219	19075	18805	19254	21486	19697	17860	19092
Poč. HELLO správ	1432	1332	1366	1296	7216	2947	1635	3023
Poč. RERR správ	1701	1619	1525	1563	1350	1577	1587	1475

50 uzlov	AODV	M	DM	TRM	EM	TM	SM	SDM
	Random-Waypoint model - max. rýchlosť 2 m/s							
PDR (%)	79.09	78.51	79.83	78.83	82.91	82.30	84.06	84.32
Oneskorenie (s)	27.605	26.524	26.499	26.570	16.206	18.324	17.848	18.987
Kontr. vs dát. správy	3.16	3.29	3.13	3.29	3.31	3.01	2.77	3.03
Poč. kontr. správ	23660	24446	23577	24566	25984	23386	21996	24201
Poč. HELLO správ	1645	1500	1499	1543	9022	4104	2006	3806
Poč. RERR správ	2070	1839	1690	1873	1451	1655	1852	1732
	Random-Waypoint model - max. rýchlosť 8 m/s							
PDR (%)	72.71	72.58	74.77	73.50	75.78	77.89	78.64	79.06
Oneskorenie (s)	38.389	46.808	38.913	39.049	24.461	30.159	28.547	27.670
Kontr. vs dát. správy	3.80	4.09	3.71	3.94	4.07	3.55	3.29	3.46
Poč. kontr. správ	26136	28041	26234	27359	29122	26164	24479	25930
Poč. HELLO správ	1232	1228	1329	1296	9023	3482	1707	3361
Poč. RERR správ	2293	2078	1824	2048	1778	1942	2069	1961
	Random-Waypoint model - max. rýchlosť 14 m/s							
PDR (%)	70.68	70.61	72.72	71.65	73.25	74.06	77.41	77.65
Oneskorenie (s)	40.217	46.854	38.801	41.401	33.924	35.585	30.077	33.880
Kontr. vs dát. správy	4.08	4.26	3.87	4.05	4.49	3.99	3.39	3.66
Poč. kontr. správ	27333	28443	26617	27473	31092	27949	24792	26859
Poč. HELLO správ	1358	1206	1282	1239	9022	3331	1666	3531
Poč. RERR správ	2452	2233	1973	2147	2023	2160	2099	2120
	Gauss-Markov model - max. rýchlosť 2 m/s							
PDR (%)	67.19	68.29	69.06	69.17	72.07	72.09	72.92	72.71
Oneskorenie (s)	70.253	66.934	62.075	63.798	43.789	54.788	54.580	54.199
Kontr. vs dát. správy	4.64	4.48	4.32	4.48	4.52	4.20	3.91	4.20
Poč. kontr. správ	29431	28788	28006	29138	30632	28522	26752	28782
Poč. HELLO správ	1219	1159	1188	1056	9024	3111	1545	2935
Poč. RERR správ	2648	2103	2007	2179	1826	2064	2164	2121
	Gauss-Markov model - max. rýchlosť 8 m/s							
PDR (%)	66.38	67.18	67.01	66.25	69.32	69.59	72.08	72.08
Oneskorenie (s)	65.809	63.924	71.247	63.155	42.957	54.435	50.965	50.876
Kontr. vs dát. správy	4.60	4.71	4.67	4.76	4.91	4.43	3.98	4.16
Poč. kontr. správ	28752	29830	29508	29823	32026	29027	27071	28278
Poč. HELLO správ	1205	1149	1085	1265	9025	3089	1511	3132
Poč. RERR správ	2607	2256	2201	2322	2041	2232	2283	2212
	Gauss-Markov model - max. rýchlosť 14 m/s							
PDR (%)	60.14	60.67	60.85	61.57	60.75	62.14	65.64	65.48
Oneskorenie (s)	80.717	91.598	72.929	73.365	63.415	69.042	67.475	69.596
Kontr. vs dát. správy	5.51	5.70	5.49	5.51	6.56	5.62	4.88	5.08
Poč. kontr. správ	31325	32619	31544	32089	37590	32910	30219	31400
Poč. HELLO správ	1013	931	952	941	9023	2544	1255	2694
Poč. RERR správ	3014	2665	2508	2532	2648	2725	2703	2641

Dodatok B

Priložené CD

Priložené CD obsahuje inštaláciu simulačného nástroja NS2 vo verzii 2.34, ako aj implementáciu jednotlivých verzií modifikovaného AODV protokolu použitých v tejto práci a simulačný skript použitý k vytvoreniu a spusteniu samotnej simulácie.